

Maier, Thomas
HTML und CSS

Graz : css4 2019, 175 S.



Quellenangabe/ Reference:

Maier, Thomas: HTML und CSS. Graz : css4 2019, 175 S. - URN: urn:nbn:de:0111-pedocs-206986 - DOI: 10.25656/01:20698

<https://nbn-resolving.org/urn:nbn:de:0111-pedocs-206986>

<https://doi.org/10.25656/01:20698>

Nutzungsbedingungen

Dieses Dokument steht unter folgender Creative Commons-Lizenz: <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.de> - Sie dürfen das Werk bzw. den Inhalt unter folgenden Bedingungen vervielfältigen, verbreiten und öffentlich zugänglich machen sowie Abwandlungen und Bearbeitungen des Werkes bzw. Inhaltes anfertigen: Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen. Dieses Werk bzw. der Inhalt darf nicht für kommerzielle Zwecke verwendet werden. Die neu entstandenen Werke bzw. Inhalte dürfen nur unter Verwendung von Lizenzbedingungen weitergegeben werden, die mit denen dieses Lizenzvertrages identisch oder vergleichbar sind. Mit der Verwendung dieses Dokuments erkennen Sie die Nutzungsbedingungen an.

Terms of use

This document is published under following Creative Commons-License:

<http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en> - You may copy, distribute and transmit, adapt or exhibit the work in the public and alter, transform or change this work as long as you attribute the work in the manner specified by the author or licensor. You are not allowed to make commercial use of the work. If you alter, transform, or change this work in any way, you may distribute the resulting work only under this or a comparable license.

By using this particular document, you accept the above-stated conditions of use.



Kontakt / Contact:

peDOCS
DIPF | Leibniz-Institut für Bildungsforschung und Bildungsinformation
Informationszentrum (IZ) Bildung
E-Mail: pedocs@dipf.de
Internet: www.pedocs.de

Mitglied der


Leibniz-Gemeinschaft

Thomas Maier

HTML und CSS

- ≡ Unterrichtsplanung
- ≡ Lernhandouts
- ≡ Übungshandouts
- ≡ pädagogischer Beipackzettel
- ≡ Zielformulierungen

HTML und CSS
von **Thomas Maier**, BEd
Zeppelinstraße 12A/7, 8055 Graz
www.css4.at | htmlundcss@css4.at
Copyright: CC Lizenz BY NC SA 2019

Inhaltsverzeichnis

<i>Allgemeine didaktische Konzeption</i>	6
Unterrichtsgrundsätze.....	6
Allgemeine didaktische Grundsätze.....	6
Unterrichtsprinzipien.....	6
Intensionen der Schüler innen.....	7
Eingangsvoraussetzungen.....	7
Handouts und Assignments.....	8
Flankierende Methoden.....	8
Raumausstattung.....	9
EDV Infrastruktur.....	10
Blendet Learning.....	11
Sozialformen.....	11
Exemplarisches Stundenbild.....	12
Grobziele und Kompetenzzuwachs.....	13
<i>Lernhandout 1.1 HTML Grundgerüst</i>	14
<i>Lernhandout 2.1 Basis Tags</i>	16
<i>Übungsblatt 2.1 Basis Tags</i>	18
<i>Lernhandout 2.2 Basis Elemente (img, a href)</i>	20
<i>Übungsblatt 2.2 Basis a href, img</i>	22
<i>Lernhandout 2.3 Best Practice</i>	24
<i>Lernhandout 3.1 Tabellen</i>	26
<i>Übungsblatt 1.1 HTML Grundgerüst</i>	28
<i>Lernhandout 3.2 HTML validieren</i>	30
<i>Lernhandout 4.1 style, span, color</i>	32
<i>Lernhandout 4.2 Rahmen</i>	34
<i>Lernhandout 4.3 Höhe und Breite</i>	36
<i>Übungsblatt 4.3 Farben, Rahmen und Größenangaben</i>	38
<i>Lernhandout 5.1 CSS im Head</i>	40
<i>Lernhandout 5.2 Schriftarten</i>	42
<i>Lernhandout 5.3 Schriftgrößen</i>	44
<i>Lernhandout 5.4 Text Dekoration</i>	46
<i>Übungsblatt 5.4 Schriften und Textdekoration</i>	48
<i>Lernhandout 6.1 Ausrichtung</i>	50
<i>Lernhandout 6.2 Abstände</i>	52
<i>Übungsblatt 6.2 Ausrichtung und Abstände</i>	54
<i>Lernhandout 6.3 Best Practice</i>	56
<i>Lernhandout 7.1 Hintergrund</i>	58
<i>Lernhandout 7.2 Hintergrund</i>	60
<i>Lernhandout 7.3 Farbverlauf</i>	62
<i>Übungsblatt 7.3 Hintergrund</i>	64
<i>Lernhandout 8.1 Positionierung</i>	66
<i>Lernhandout 8.2 display-Eigenschaft</i>	68
<i>Übungsblatt 8.2 Positionierung und Display</i>	70
<i>Lernhandout 8.3 Strukturierung</i>	72
<i>Lernhandout 8.4 Float</i>	74
<i>Lernhandout 9.1 Pseudoklassen (Buttons)</i>	76

Lernhandout 9.2 Pseudoklassen.....	78
Übungsblatt 9.2 Pseudoklassen.....	80
Lernhandout 9.3 iframe.....	82
Übungsblatt 9.3 iframe.....	84
Lernhandout 10.2 Pseudoelemente.....	86
Übungsblatt 10.2 Pseudoelemente.....	88
Lernhandout 10.3 Audio.....	90
Übungsblatt 10.3 Audio.....	92
Lernhandout 10.4 Video.....	94
Übungsblatt 10.4 Video.....	96
Lernhandout 11.1 Medienabfrage.....	98
Übungsblatt 11.1 Medienabfragen.....	100
Lernhandout 11.2 Objekte.....	102
Übungsblatt 11.2 Objekte.....	104
Lernhandout 11.3 Viewport.....	106
Übungsblatt 11.3 Viewport.....	108
Lernhandout 11.4 Box Eigenschaften.....	110
Lernhandout 12.1 Formulare.....	112
Lernhandout 12.2 Textarea.....	114
Lernhandout 12.3 Input.....	116
Übungsblatt 12.3 Input.....	118
Lernhandout 12.4 Input Typen.....	120
Übungsblatt 12.4 Input Typen.....	122
Lernhandout 12.5 Form Button.....	124
Übungsblatt 12.5 Formular Buttons.....	126
Lernhandout 12.6 Dialogeingaben.....	128
Übungsblatt 12.6 Dialoge.....	130
Lernhandout 12.7 Auswahleingaben.....	132
Übungsblatt 12.7 Auswahleingabe.....	134
Lernhandout 12.8 Auswahllisten.....	136
Übungsblatt 12.8 Auswahllisten.....	138
Lernhandout 13.1 Animationen.....	140
Lernhandout 13.2 Animationen.....	142
Lernhandout 13.3 Animationen.....	144
Übungsblatt 13.3 Animationen.....	146
Lernhandout 13.4 Transformieren.....	148
Übungsblatt 13.4 Transformieren.....	150
Lernhandout 13.5 Transformieren.....	152
Lernhandout 13.6 Transitions.....	154
Übungsblatt 13.6 Transitions.....	156
Lernhandout 14.1 CSS Variablen.....	158
Lernhandout 14.2 calc() Funktion.....	160
Lernhandout 14.3 counter() Funktion.....	162
Übungsblatt 14.3 CSS Funktionen.....	164

<i>Lernhandout 14.4 Attributselektoren</i>	<i>166</i>
<i>Lernhandout 15.1 Image Map</i>	<i>168</i>
<i>Übungsblatt 15.1 Image Map</i>	<i>170</i>
<i>Pädagogischer Beipackzettel</i>	<i>172</i>
<i>Warum überhaupt noch HTML?</i>	<i>172</i>
<i>Leistungsbewertung</i>	<i>172</i>
<i>Blendet Learning</i>	<i>172</i>
<i>Sozialformen</i>	<i>172</i>
<i>Lehrplanbezug</i>	<i>173</i>
<i>Die kritische Sozialformierung</i>	<i>173</i>
<i>Quellenverzeichnis</i>	<i>174</i>
<i>Internetquellen und online Tools</i>	<i>174</i>
<i>Literaturverzeichnis</i>	<i>174</i>

Allgemeine didaktische Konzeption

Unterrichtsgrundsätze

Im Unterricht soll ein Klima der Wertschätzung sowie des Respekts und der Gleichbehandlung vorherrschen.

Das Klassenzimmer soll ein Ort der offenen Kommunikation sein. Der Unterricht ein Forum für Fragen und Diskussionen. Fragen aller Art sind erwünscht. Getreu: "Es gibt keine dummen Fragen – die einzige dumme Frage ist jene, die nicht gestellt wird".

Im Unterricht sollte Zeit für individuelles Fördern gefunden werden. Leistungsschwächere Schüler|innen sollen gefördert werden. Leistungsstärker Schüler|innen gefordert. Dabei sollte dem "Schüler-Schüler-Coaching" Vorrang gegeben werden.

Der Fokus liegt auf den Leistungen der Schüler|innen. Die besondere Würdigung von Qualitäten soll einer fehlerzentrierten Wahrnehmung Vorzug gegeben werden. Fehler machen ist erlaubt – jedoch sollen diese kommuniziert werden.

Die Leistungen der Schüler|innen sind sowohl auf der Prozessebene als auch auf der Produktebene zu würdigen. Es wird nicht nur das aktive Lernen gefordert, sondern auch das passive Lernen akzeptiert.

Allgemeine didaktische Grundsätze

- 📖 Unterrichtsbeispiele mit Praxis- und Lebensbezug.
- 📖 Methodenvielfalt und kooperatives offenes Lernen
- 📖 Förderung von Selbstständigkeit, Selbstverantwortung, Einzel- und Teamarbeit, sozialem Lernen.
- 📖 Einsatz von Blended Learning (IT-Bezug), Standardsprache und Fachterminologie
- 📖 Individuelles Fördern bei einer Maximierung der reinen Lernzeit.
- 📖 Handlungsorientierte Unterrichtsplanung und Einhaltung der Progressionslogik
- 📖 Produkt und Prozessorientierte Leistungsbewertung

Unterrichtsprinzipien

Es gelten die allgemeinen Unterrichtsprinzipien laut dem Bundesministerium für Bildung. Die fettgedruckten Prinzipien wurde in den Detailplanungen Rechnung getragen. Andere sollten unbedingt im Rahmen des pädagogischen/situationsgemäßen Gesprächs vermittelt werden. Vergessen Sie niemals diese äußerst wichtigen Prinzipien – sie sind weit mehr als einfaches fächerübergreifendes Lernen/Lehren.

- Erziehung zur Gleichstellung von Frauen und Männern
- **Gesundheitserziehung**
- Interkulturelles Lernen
- **Leseerziehung**
- **Medienbildung**
- Politische Bildung
- Sexualerziehung
- **Umweltbildung**
- Verkehrserziehung
- Wirtschaftserziehung und Verbraucher/innenbildung

Intensionen der Schüler|innen

Die Schüler|innen wollen ...

- 📄 grundlegende und vertiefende Kenntnisse des Webdesigns erwerben.
- 📄 eigene Webseiten nach eigenen Vorstellungen entwerfen.
- 📄 das Internet mitgestalten.

Eingangsvoraussetzungen

Das Erlernen von HTML und CSS kann nicht sofort ohne weiteres starten. Es sind Kompetenzen wie das Englische auf Niveau A2 sowie das Schreiben mit 10-Fingern klar von Vorteil. Viel wichtiger ist, grundsätzliche Erkenntnisse der Typographie, Design und Layout, Gestaltungslehre und einem (wohl am wichtigsten) Ordnungsbewusstsein als Eingangskompetenzen anzuführen. Da nun also, HTML und CSS als eine Vertiefung der konzeptionellen Erkenntnisse der Textverarbeitung (in primärer Hinsicht) darstellt, ist es erst nach der Kenntnisvermittlung einer allgemeinen Textverarbeitung als diskussionswürdig darzustellen.

Es stellen sich die kognitiven Probleme im einen Feld genauso dar, wie sie es im wechselseitig anderem tun. So will die Darstellung von Text sowohl im naheliegenden Einfachen die Entwicklungsmöglichkeiten für die|den Lernende|n dieselben Erfahrungsmomente nach sich ziehen.

Exemplarisch: Die Unterscheidung zwischen einer Zeilenschaltung und einer Absatzschaltung in MS Word, liegt im Drücken der Shift-Taste. Die Kennzeichnung einer Zeilenschaltung in HTML bedarf eines `
` Tags – und noch extremer: die Absatzschaltung wird durch einen Start-Tag `<p>` und End-Tag `</p>` realisiert.

So bleiben also, jene Textverarbeitungslehrer|innen weiterhin die einführende Instanz und sollten aufgrund ihrer Kompetenzen niemals als vernachlässigt betrachtet werden. Sie sind es, die einen Text die Lesbarkeit verleihen – neben der äußerst wichtigen Arbeit von Germanisten (im Deutsch-Unterricht) und im besonderen (Kunstlehrer|innen) die den Spass an der textuellen Veränderung anstreben um damit den Nimbus der kreativen textuellen Gestaltung zu nähren.

Das Verständnis eines Textes ist nun also abhängig von a) der Grammatik und seiner Orthographie, b) seinem Inhalt und Aussage – seines philosophischen Wesens und c) der Darstellung – der Textverarbeitung – seines Designs und Layout (so wie die lateinischen Zeichen aneinandergereiht sind).

Handouts und Assignments

Die Handouts sind in Einheiten gegliedert. Jede Einheit beschreibt einen Themenkomplex aus dem jeweiligen Bezugssystem. Grundsätzlich unterliegen die Einheiten einer Progressionslogik, können aber auch voneinander unabhängig unterrichtet werden (je nach pädagogischen Bedarf).

Alle Handouts wurden nach dem Prinzip des "One-Page-Management" erstellt und können auch als Kopiervorlage genutzt werden. Die Handouts wurden mit Screenshots bebildert. Die Anleitungen sind auf die Übungsbeispiele hin konzipiert.












Die Symbole auf den Handouts sollen den Schüler|innen eine Ordnung im Lernprozess vermitteln.

	↩ Übungsbeispiel
	↩ Information und Hervorhebung
	↩ Nachschlagen und Bibliothek
	↩ Tools und Werkzeuge
	↩ Notizen und Anmerkungen

Dieses One-Page-Management wurde der Wirtschaft entnommen. Es ist eine Methode um Führungspersonlichkeiten einen raschen Überblick über die aktuellen Fakten zu geben. Rasch und Einfach – was nun also für Manager von großen Betrieben stimmig ist, sollte für den Schüler bzw. die Schülerin genauso gelten. Ein neues Thema – ein neues Blatt Papier.

Flankierende Methoden

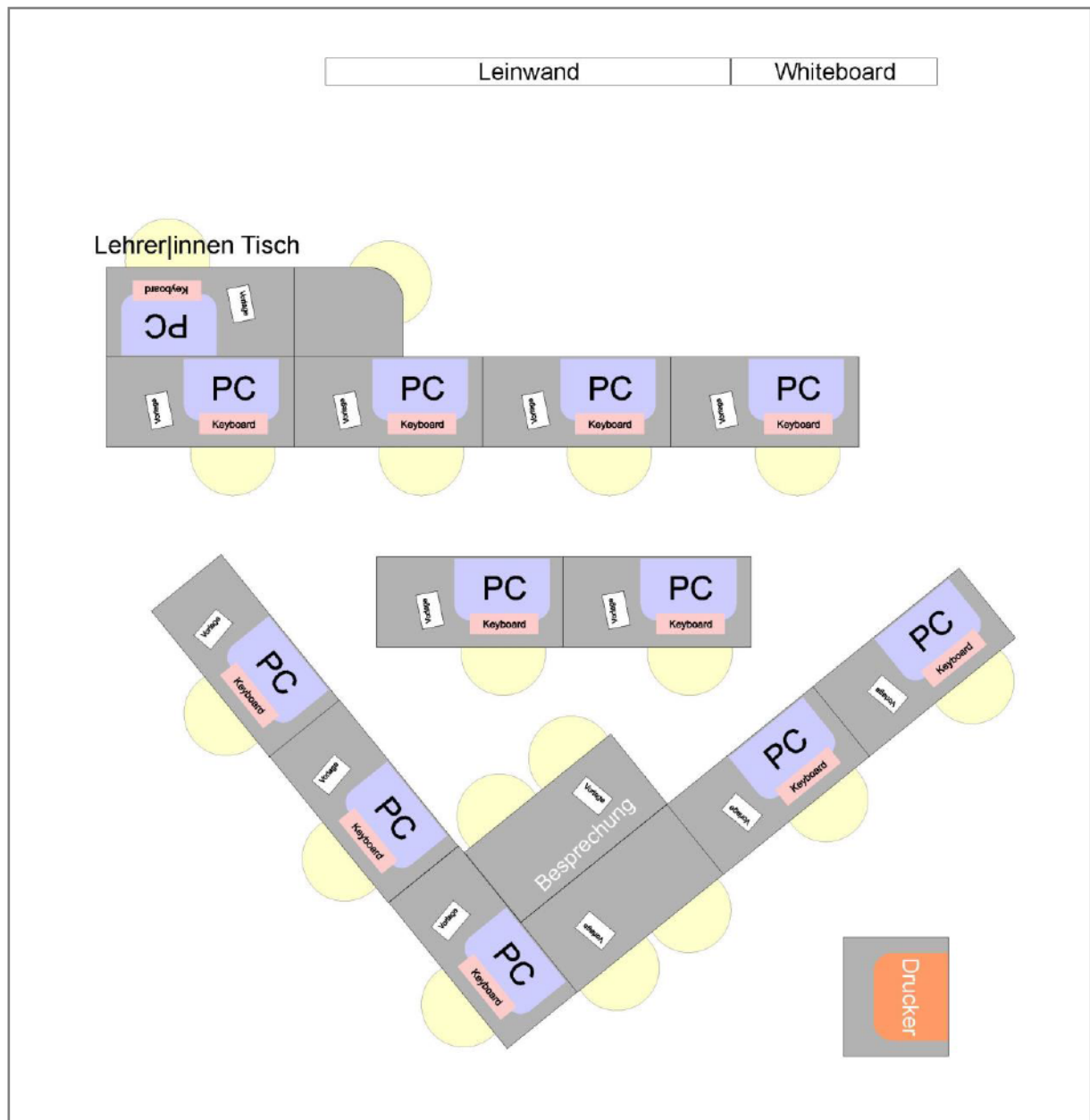
Neben den klassischen Methoden Frontalunterricht, Erklären, Lehrervortrag und Vorzeigen-Nachmachen, sind folgende Unterrichtsmethoden ebenfalls denkbar:

-  Aufgabenorientiertes Lernen
-  Fächerübergreifendes und –verbindendes Lernen
-  Handlungsorientiertes Lernen
-  Einzelarbeit
-  Konstruktionsaufgaben
-  Projektlernen
-  Freie Arbeit
-  Entdeckendes Lernen
-  Mind-Mapping
-  Brainwriting
-  uvm.

Raumausstattung

Folgende Medien und Ausstattungen sind für den Unterricht empfehlenswert:

- Großes Whiteboard
- Beamer (Projektor)
- Besprechungstische
- ausreichend Bürostühle
- jeweils einen Bürotisch pro PC (Arbeitsplatz – PC sind rechts angeordnet, damit die Linke Hälfte des Tisches Platz für Vorlagen bietet).
- ein Drucker



Vorschlag für eine Raumgestaltung

EDV Infrastruktur

HTML und CSS ist äußerst genügsam was die Leistung der Hard- und Software betrifft. Im Grunde reicht ein gängiger PC mit einem Editor und einem Browser. Das Betriebssystem spielt nur eine untergeordnete Rolle, da HTML und CSS plattformunabhängig ist. Ob nun Windows, OSX oder Linux – egal!

Dennoch sind folgende Software-Spezifikationen empfehlenswert (ein {+} steht für zwingend notwendig und ein {-} für vernachlässigbar:

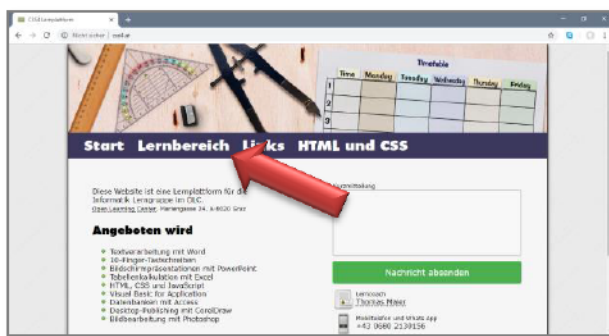
- Notepad ++ (Texteditor) {+}
- Adobe Dreamweaver oder Microsoft Expression Web (HTML Editor)
- Adobe Photoshop oder Gimp (Bildbearbeitung) {+}
- sämtliche gängige Browser (Chrome, Firefox, Edge usw.) {+}
- Audacity (Audioeditor)
- Microsoft Word oder OpenOffice (Textverarbeitung)
- CMS Editoren (z. B. Adobe Muse, Joomla)
- weitere Tools: Color Picker, ZIP Software, Screenshot-Tools, PDF Viewer (Acrobat odgl.) {+}
- kolaborative Anwendungen: Office 365, OneNote, Google, TeamViewer {-}
- didaktische Tools: iTalc, Moodle
- FTP Software (z. B. FileZilla) {-}
- Virtualisierungssoftware {-}
- SQL Server und PHP Sandbox
- Entwicklungsumgebungen (z. B. Visual Studios) {-}

Eine Webanbindung des Schulservers (z. B. über IIS, Apache odgl.) wäre sicher ein Pluspunkt, damit die Schüler|innen ihre Webseiten sofort online stellen können. Die Planung eines Webserver ist aber ein anderes Thema, das den Rahmen dieser Arbeit sprengen würde. Die Schüler|innen können aber ihre Lernprodukte auf gratis Webspaces anbieten, wie z. B. mur.at oder Lima-City.de publizieren (wenn nötig und wenn sie es wollen).

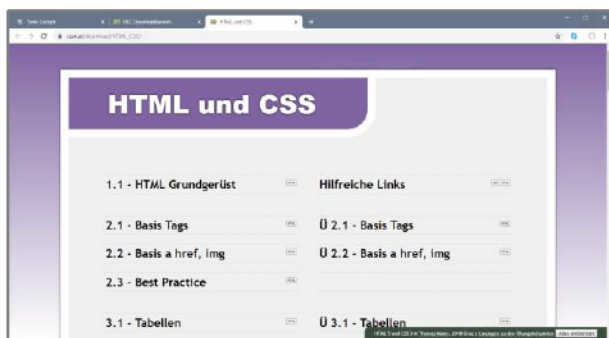
Die Hardware ist definitiv genügsamer:

- Bei Laptops muss ein externes Keyboard bereitgestellt werden. Ebenso eine Maus.
- Ein zweiter Monitor ist ein Nice-to-Have, aber nicht zwingend notwendig.
- Eine Breitband-Internet-Verbindung.
- Ein funktionierendes Netzwerk mit Serverseitigen Profilen und Netzlaufwerken.
- Audio-Ausgabe: Entweder über Lautsprecher oder besser über Kopfhörer.
- USB Anschlüsse für die Verwendung von USB-Speicher-Sticks.
- Ein Netzwerk-Drucker

Blendet Learning



Auf der Lernplattform www.css4.at findet man einen Link zum Lernbereich, wo man die Lernsite für HTML und CSS aufrufen kann.



Die Lernplattform für HTML5 und CSS3 erlaubt zu jeder Einheit die Downloads der Lern- und Übungshandouts und der benötigten Dateien für die Übungsbeispiele. Zusätzlich findet man auf der Lernplattform noch hilfreiche Links zum Unterricht. Links findet man die Lernhandouts und korrespondierend rechts die dazu passenden Übungshandouts.

Zu allen Übungsbeispielen gibt es selbstverständlich eine Lösung, die in den meisten Fällen nur als Referenz dient und nicht als absolute Lösung (ohne Abweichung von Komma und Beistrich). Die Lösungen können über die eMail-Adresse webmail@css4.at geordert werden. Geben Sie dafür Ihre eMail-Adresse und einen Link auf die Schulhomepage (zur Identifikation Ihrer Person als Lehrer|in) bekannt. Die Verwendung von Lern- und Übungshandouts, sowie der Lösungen ist selbstverständlich für Lehrer|innen von öffentlichen Schulen oder von Organisationen mit öffentlichen Charakter kostenlos und frei verwendbar.

Sollten Sie das Unterrichtsmaterial von www.css4.at kommerziell verwenden (z. B. für bezahlte Kurse in der Erwachsenenbildung) dann bitte ich Sie recht höflich mir zuvor eine Mail an webmail@css4.at mit dem didaktischen Umfang ihres Unterrichts (didaktische Konzeption) und dem damit verbunden finanziellen Volumen zuzusenden. Wir finden sicher eine Modus für die Verwendung des Unterrichtsmaterials.

Sozialformen

Das Unterrichtsmaterial ist auf Einzelarbeit ausgelegt. Dennoch ist Partnerarbeit bei Bildschirmknappheit denkbar. Gruppenarbeit kommt kategorisch nicht vor. Das Plenum wird nur im Rahmen des Frontalunterrichts aktiviert.

Methoden, die andere Sozialformen bedingen, unterliegen der didaktischen Kreativität des|der Lehrerin. Gegen Ende des Unterrichts kann gerne eine Projektarbeit angesetzt werden, die auch in der Gruppe gelöst werden kann – eine daraus resultierende Sozialkompetenz ist aber ehre nicht zu erwarten.

Exemplarisches Stundenbild

↕ Phase	Dauer	Wann? Zeit bis	Was? Lerninhalte	Wie? Methoden	Wodurch? Medien	Warum? Methode
Einstieg	2 min	2'	Begrüßung	päd. Gespräch	---	---
	5 min	7'	Vorbesprechung	päd. Gespräch	---	Erhebung von Interessen
	6 min	13'	Wiederholung	Mind-Mapping	A3 Bogen oder Whiteboard	Wiederholung der letzten UE.
Erarbeitung	10 min	23'	Einheit x	Vorzeigen – Nachmachen	Bildschirm	Vermittlung der Skills
	30 min	53'	Übungsbeispiele Einheit x	freies Arbeiten	Bildschirm	Kombiniert mit individuellen Coaching
	5 min	58'	PAUSE			
	10 min	68'	Einheit y	Vorzeigen Nachmachen	Bildschirm	Vermittlung der Skills
	30 min	98'	Übungsbeispiele Einheit y	freies Arbeiten	Bildschirm	Kombiniert mit individuellen Coaching
	15 min	113'	Individuelles Coaching und freies Arbeiten	Coaching	Bildschirm	Leistungsbeobachtung, Weiterführende Fragen beantworten
Erfolgssicherung	5 min	118'	Ausblick auf die nächste UE.	päd. Gespräch	---	Die nachgelagerten Themen ansprechen
	2 min	120'	Verabschiedung	päd. Gespräch	---	Verabschiedung und Herstellung der Arbeitsplatz-Ordnung. (z. B. Herunterfahren des PC)

Grobziele und Kompetenzzuwachs

Der die Schüler in ...	Zielart	Taxonomie
... erlernen die Verwendung von HTML und CSS.	I	I
... nutzen das Wissen zur Gestaltung von Webseiten.	I	II
... erhalten das Rüstwerkzeug um individuelle Vorstellung vom Webdesign (jenseits von Vorlagen und Templates) umzusetzen.	SIKA	II
... erhalten einen Einblick in die darunterliegende Ebene der IT-konformen Darstellung und betrachten den kritischen letzten Eindruck einer Darstellung.	KI	III
... nutzen die erlernten Fähigkeiten um visuelle Vorstellungen und Gedanken in schriftlich/auditiver Form darzustellen.	KI	IV
... entwickeln ein höheres Professionsbewusstsein.	ASK	IV
... erwirbt die Fähigkeit selbstständig sein ihr Wissen zu erweitern.	SIK	III
... ist befähigt ein mittleres und statisches Web-Projekt zu erledigen.	SI	IV
... steigert seine ihrer kognitive Leistungen im Bereich des linearen und logischen Denken.	KS	IV
... entwickelt Problemlösungsstrategien.	KS	IV
... steigert sein ihr Bewusstsein für Ordnung und Genauigkeit.	AS	IV

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Lernhandout 1.1 HTML Grundgerüst

Referenzcode: HCL011

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... verstehen die Funktion von Start und End-Tags	K	I
... erkennen die Struktur des HTML Grundkonstrukts.	K	I
... schreiben eine Basis HTML Seite (Grundkonstrukt).	I	II
... kennen die Elemente des Grundkonstrukts.	K	III
... speichern HTML Seiten mit richtigen Namen und Dateierweiterungen	I	II
... erstellen eine Vorlage für spätere Arbeiten.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Ein HTML-Tag beginnt mit einem Start-Tag und endet mit einem End-Tag. Dazwischen ist der zu formatierende Bereich (z. B. Text).

Das Grundkonstrukt jeder HTML-Seite beginnt mit einem `<html>` Tag. Darin befindet sich der `<head>` und der `<body>`. Im `<head>` stehen Metadaten wie z. B. der Titel. Im `<body>` ist der Inhalt der Website, der dann im Browser angezeigt wird.

```
<!doctype html>
<html lang="de">
  <head>
    <meta charset="utf-8" />
    <title>Titel der Webseite</title>
    <meta name="author" content="John Doe">
    <meta name="date" content="2018-04-20">
  </head>
  <body>
    Hier ist der sichtbare Teil der Website!
  </body>
</html>
```



`<!doctype html>` → Definiert den Code als HTML Seite
`<html lang="de">` → HTML-Starttag und als Sprache ist Deutsch
`<head>` → Head-Starttag
`<meta charset="utf-8" />` → UTF8 Zeichenkodierung (ä, ö, ß usw.)
`<title>Titel der Webseite</title>` → Der Titel der Website
`<meta name="author" content="John Doe">` → Name des Autors
`<meta name="date" content="2018-04-20">` → Datum
`</head>` → Head-Endtag
`<body>` → Body-Starttag
`</body>` → Body-Endtag
`</html>` → HTML-Endtag



Speichern unter ...

HTML Dateien haben die Extension (Dateierweiterung) `*.htm` bzw. `*.html`. Die `index.html` wird angezeigt, wenn man im Browser die URL eines Ordners eingibt – z. B: `www.johndoe.com/projekte`. Das ist aber abhängig von den Einstellungen des Webserver – manchmal kann es auch `default.htm`, `start.htm`, `index.htm` lauten!



Übung A: Das HTML Grundkonstrukt

- ☐ Starte einen Webeditor (Notepad ++, Dreamweaver odgl.)
- ☐ Erstelle ein HTML Grundgerüst.
- ☐ Trage deinen Namen als Autor ein
- ☐ Speichere es als `vorlage.html` ab!



Wir arbeiten in Zukunft immer mit der `vorlage.html` – speichere sie also so ab, dass du immer Zugriff darauf hast (USB-Stick, Cloud, Netzlaufwerk).

Lernhandout 2.1 Basis Tags

Referenzcode: HCL021

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... verwenden Tags richtig (öffnen und schließen von Tags).	I	II
... definieren Überschriften <h1> bis <h6>	I	II
... kennen Standalone Tags wie z. B. <hr>	K	I
... kennen den Unterschied zwischen und <p> Elementen	K	I
... führen div. HTML Zeichenformatierungen durch.	I	II
... können Tags verschachteln und kennen das Prinzip von Eltern- und Kindelementen.	K	II
... erstellen HTML Listen und Aufzählungen. (Elternelement: Kindelement:).	I	II
... wissen, was Attribute sind und nutzen diese für Listen und Aufzählungen.	I	II
... validieren ihren HTML Code online über w3.org	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- 📄 In der Anfangsphase wird man noch zwischen dem Editor und der Browserdarstellung wechseln um die Auswirkungen sofort zu sehen. → F5
- 📄 Für Tieferstellungen <sub> kann man eine chemische Formel vorzeigen oder entwickeln lassen. z. B: H_2SO_4 → `<p>H₂SO₄</p>`
- 📄 Für Höherstellungen <sup> kann man den Pythagoras vorzeigen oder entwickeln lassen. z. B: $a^2 + b^2 = c^2$
`<p>a² + b² = c²</p>`
- 📄 Mit dem <button> Tag soll plastisch gezeigt werden, das es sich um ein Element handelt.

HTML-Tags zur Zeichen und Absatzformatierung

Hier die wichtigsten HTML-Tags. Nicht vergessen: Jeder Tag muss geschlossen werden! Ausnahmen bilden die Standalone-Tags (z. B. `
` oder `<hr>`).

<code><h1> Überschrift der 1. Ordnung </h1></code> <code><h6> Überschrift der 6. Ordnung </h6></code>	<code>
</code> eine Zeilenschaltung <code><hr></code> horizontale Linie
<code><p> Absatz </p></code>	Tags können auch verschachtelt werden. Dann spricht man von einem Eltern- und einem Kindelement. <code><p>Das hier ist <i>besonders Wichtig</i> und muss hervorgehoben werden!</p></code> Im Browser wird folgendes dargestellt: Das hier ist <i>besonders Wichtig</i> und muss hervorgehoben werden!
<code><i> kursive Schrift </i></code> <code> Fettschrift </code> <code><sub> tiefgestellt </sub></code> <code><sup> hochgestellt </sup></code> <code><s> durchgestrichen </s></code>	
<code><button> Hier klicken </button></code>	

Listen und Aufzählungen

Nicht nummerierte Liste (unordered list)		Nummerierte Liste (ordered list)	
Beispiel: <ul style="list-style-type: none"> ○ HTML lernen ○ HTML anwenden ○ CSS lernen 		Beispiel: <ol style="list-style-type: none"> 1. HTML lernen 2. HTML anwenden 3. CSS lernen 	
<code></code> <code> ... </code> <code> ... </code> <code></code>	<code></code> bedeutet List-Item. Also der Aufzählungspunkt.	<code></code> <code> .. </code> <code> .. </code> <code></code>	<code></code> bedeutet List-Item. In diesem Fall die Nummerierung.
<code><ul type="circle"></code> Aufzählung mit nicht ausgefüllten Kreisen <code><ul type="square"></code> Aufzählung mit Rechtecken Achtung: Das Type-Attribut im <code></code> Tag wird unter HTML5 nicht mehr unterstützt. Die meisten Browser zeigen es aber trotzdem an. Nur bei einer Validierung des HTML-Codes kommt der Hinweis auf CSS.		<code><ol type="I"></code> Nummerierte Liste mit Römischen Zeichen. Weitere Werte für das type-Attribut: 1 ... Arabische Zahlen A ... Großbuchstaben a ... Kleinbuchstaben i ... kleine Römische Zeichen <code><ol start="3"></code> Startet die nummerierte Liste bei Drei. <code><ol reversed></code> Die Reihenfolge wird umgedreht!	



Bei `type=""` im `` oder `` Tag handelt es sich um ein Attribut. Attribute bestimmen im HTML-Element zusätzliche Eigenschaften/Informationen. Sie stehen immer im Starttag und werden in der Regel als `name = "value"` verwendet. (Name ist die Bezeichnung des Attributs, Value ist der Wert)



Web-Tipp

<http://validator.w3.org> ← den eigenen HTML Code auf Fehler überprüfen

Übungsblatt 2.1 Basis Tags

Referenzcode: HCU021

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... verwandeln ein TXT File in ein HTML Dokument	I	II
... arbeiten mit dem HTML Grundkonstrukt	I	II
... führen div. Elementzuweisungen und Zeichenformatierungen durch	I	II
... erstellen Listen und Aufzählungen	I	II
... begeistern sich für strukturiertes Arbeiten.	A	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- 📄 Die Datei **LW_Salzburg_2013.txt** muss zum Download bereitgestellt werden!
Siehe www.css4.at → HTML und CSS



Übung A: TXT in HTML

Öffne die Datei "LW_Salzburg_2013.txt" mit einem Texteditor und speichere diese als "LW_Salzburg_2013.html" ab.

- ❑ Füge um den Text das HTML-Grundkonstrukt ein.

```
<!doctype html>
<html lang="de">
  <head>
    <meta charset="utf-8" />
    <title>Landtagswahl Salzburg 2013</title>
    <meta name="autor" content="Vorname Nachname">
    <meta name="date" content="jjjj-mm-tt">
  </head>
  <body>
    Der Text
  </body>
</html>
```

- ❑ Weise den Überschriften die passenden Tags zu. <h1> bis <h3>
- ❑ Verteile <p> Tags im Text um Absätze zu bestimmen.
- ❑ Die Aufzählung mit den Parteien soll als "nicht nummerierte Liste" (unordered list) dargestellt werden. und

Folgende Parteien traten zur Wahl an:

- BZÖ - Bündnis Zukunft Österreich
- FPÖ - Freiheitliche Partei
- Grünen - Die Alternativen
- KPÖ - Kommunistische Partei Österreich
- ÖVP - Österreichische Volkspartei
- PIRAT - Piratenpartei
- SPÖ - Sozialdemokraten
- TS - Team Stronach

- ❑ Die Liste mit den Verteilungen der Sitze soll als "nummerierte Liste" (ordered list) dargestellt werden! und

Verteilung der Sitze

Nach der Wahl wurden die Sitze im Landtag wie folgt verteilt:

1. ÖVP - 11 Sitze (Schwarz)
2. SPÖ - 9 Sitze (Rot)
3. Grüne - 7 Sitze (Grün)
4. FPÖ - 6 Sitze (Blau)
5. TS - 2 Sitze (Gelb)

- ❑ Alle Vor- und Nachnamen in dem Text sollen kursiv dargestellt werden. <i>
- ❑ Das Wort "Salzburg" soll immer fett geschrieben werden.

Lernhandout 2.2 Basis Elemente (img, a href)

Referenzcode: HCL022

Feinziele Die Schüler innen ...	Zielart	Taxonomie
... binden Hyperlinks ein.	I	II
... verstehen und nutzen das href Attribut	I	III
... verstehen und nutzen das target Attribut	I	III
... binden Bilder in den HTML Code ein.	I	II
... verstehen und nutzen die Attribute des Elements.	I	III
... möchten das Internet barrierefreier mitgestalten (alt Attribut des).	A	II
... kennen Bildtypen und differenzieren durch ihre Eigenschaften.	K	III

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- 📄 Bilder für "Vorzeigen und Nachmachen" müssen selbst organisiert werden.
- 📄 Urheberrechte von Bildern beachten.
- 📄 Das alt Attribut sollte unbedingt erklärt werden und seine Bedeutung für das barrierefreie Web. (Stichwort: Screen Reader für Menschen mit Sehbehinderungen).

Hyperlinks

HTML	<code> ... </code>
------	--

<code>Google Suche</code>



Links werden mit dem `<a>` Tag und dem `href` Attribut definiert.

`href=""` Innerhalb der Anführungszeichen wird der Verweis eingetragen.
 Ein absoluter Link ist ein Hyperlink zu einer fixen Adresse
 z. B. `"https://www.google.at"`
 Der relative Link verweist abhängig vom Speicherort
 z. B.: `"../html/grundbegriff.html"`
`"mailto: john.doe@google.at"` ← mit E-Mail-Client öffnen
`"skype: johndoe90"` ← für Skype
`"tel:+4301511452"` ← für Telefonnummern

`target=""` Legt das Zielfenster für die Verweise fest.
`_self` öffnet im aktuellen Fenster (Standardwert)
`_blank` öffnet in neuem Fenster.
`_parent` Elternfenster
`_top` oberstes Fenster

Bilder

HTML	<code></code>
------	--

<code></code>
--



Bilder werden mit dem `` Tag eingefügt. Es handelt sich dabei um ein Standalone-Tag ohne Elementinhalt und ohne End-Tag.

`src=""` Verweis auf den Speicherort des Bildes.
 (relativ oder absolut)

`alt=""` Beschriftung des Bildes.

`title=""` Zeigt einen Tooltip, wenn man mit der Maus darüber fährt.

`height=""` Höhe des Bildes (in Pixel oder Prozent) ohne einer Angabe wird die Originalhöhe des Bildes verwendet.

`width=""` Breite des Bildes (in Pixel oder Prozent) ohne einer Angabe wird die Originalbreite des Bildes verwendet.



Ein Hyperlink kann auch auf ein Bild gelegt werden:
``



Die wichtigsten Bildtypen mit Eigenschaften

- *.gif → 256 Farben, geringer Speicherbedarf, transparenter Hintergrund (eine Farbe transparent schalten), animierbar.
- *.png → hohe Qualität mit transparenten Hintergrund
- *.jpg → sehr hohe Qualität, stark komprimierbar – dadurch eine geringe Dateigröße, keine Transparents
- *.svg → für Vektorgraphiken, frei skalierbar ohne Qualitätsverlust.

Übungsblatt 2.2 Basis a href, img

Referenzcode: HCU022

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... erstellen ein Inhaltsverzeichnis mit HTML.	I	II
... verschachteln und kombinieren von Listen und Aufzählungen.	K	II
Übung B		
... erstellen eine Top-Ten Aufstellung mit HTML.	I	II
... reflektieren auf ihre Lieblingsseiten (bevorzugte) im Internet.	A	III
Übung C		
... binden Bilder in den HTML Code ein.	I	II
... kombinieren Bilder mit Hyperlinks.	I	II
... nutzen Bilder um zu emotionalisieren (in Verbindung mit Hyperlinks als Button).	A	II
... kennen mind. vier Suchmaschinen.	K	I

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- ☞ Durch die Screenshots auf dem Übungsblatt sollen die Schüler|innen das Endprodukt antizipieren.



Übung A: Ein Inhaltsverzeichnis

- ☐ Erstelle das HTML-Grundgerüst (Vorlage.html) mit deinem Namen und dem aktuellen Datum. Der Titel ist Inhaltsverzeichnis.
- ☐ Schreibe ein Inhaltsverzeichnis! (wie neben dargestellt).
- ☐ Die Listen müssen verschachtelt sein!
- ☐ Du benötigst folgende Tags: `<h1>`, ``, ``, ``, ``, `<s>`, `
`
- ☐ Als HTML-Dokument speichern!

Inhaltsverzeichnis

- I. **Einleitung**
Einige Worte zur Begrüßung
- II. **Projektbeschreibung**
Ein kurzes Abstract über das Projekt
- III. **Projektplanung**
 - a. Ziele setzen
 - b. Soll-Ist-Vergleich
 - c. Ziele überprüfen
- IV. **Realisierung**
Das Projekt wird ausgeführt
- V. **Checkliste**
 - Kopieren
 - Verteilen
 - Erklären
 - Abhacken
- VI. **Risikomanagement**



Übung B: Top-Ten Websites

- ☐ Erstelle das HTML-Grundgerüst (Vorlage.html) mit deinem Namen und dem aktuellen Datum. Der Titel ist Top-Ten.
- ☐ Schreibe eine "Top-Ten Liste" von beliebten Webseiten (wie neben dargestellt).
- ☐ Verwende hier Hyperlinks zu den Webseiten!
- ☐ Vollende die Liste bis Platz 1 mit deinen Lieblingsseiten im Netz!
- ☐ Du benötigst folgende Tags: `<h1>`, ``, ``, `<a href>`, `<s>`, `
`
- ☐ Als HTML-Dokument speichern!

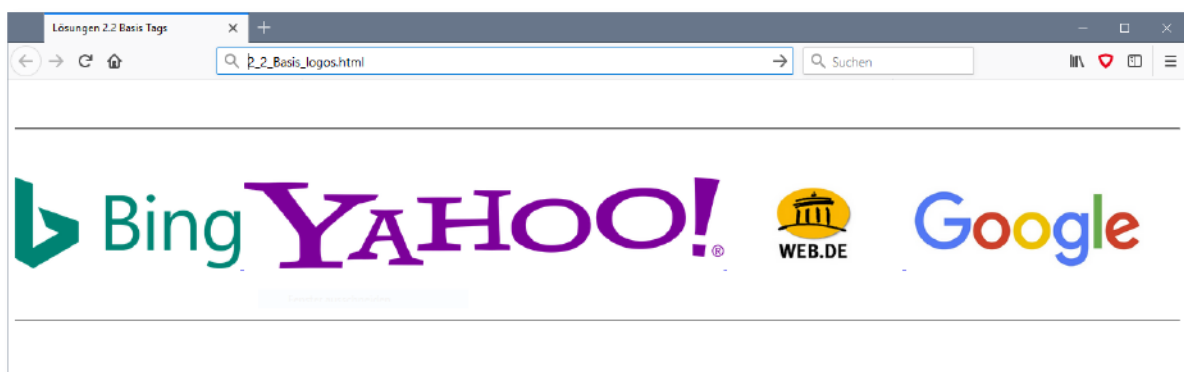
Top-Ten Websites

10. Selfhtml.org
9. HTML Code validieren
8. Google Suche
7. Facebook
6. ...



Übung C: Suchmaschinen

- ☐ Erstelle das HTML-Grundgerüst (Vorlage.html) mit deinem Namen und dem aktuellen Datum. Der Titel ist Suchmaschinen.
- ☐ Suche im Internet nach den Logos von mind. vier Suchmaschinen.
- ☐ Die Logos sollen nebeneinander dargestellt werden (siehe unten).
- ☐ Mit einem Klick auf das Logo soll sich die jeweilige Suchmaschine in einem neuen Browserfenster öffnen.
- ☐ Du benötigst folgende Tags: ``, `<a href>`
- ☐ Als HTML-Dokument speichern!



Lernhandout 2.3 Best Practice

Referenzcode: HCL023

Feinziele Die Schüler innen ...	Zielart	Taxonomie
... strukturieren den HTML Code lesbarer.	I	II
... haben ein gesteigertes Ordnungsbewusstsein.	A	II
... erkennen die Struktur von Kind und Elternelementen.	K	I
... begreifen, dass es zu Problemen kommen kann, wenn man HTML-eigne Zeichen für den darstellbaren Text verwendet.	K	I
... nutzen die HTML Zeichenreferenz.	I	II
... verwenden HTML Kommentare um Anmerkungen im Code zu platzieren.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Dem Browser ist es relativ egal wie man den Code strukturiert. Zeilen- und Absatzschaltungen werden im Browser nicht dargestellt – dafür verwendet man `
` oder `<p>`. Man kann theoretisch auch eine Website in nur einer Zeile schreiben:

```
<body><div>Aufzählung<ul><li>Punkt 1</li><li>Punkt 2</li></ul></div></body>
```

Um nun aber den Code für den Menschen lesbarer zu machen, rückt man die Child-Elemente (Kindelemente) mit Leerzeichen ein:

<code><body></code>	← Parent (Elternelement)
<code> <div>Aufzählung</code>	← First Child (Kindelement) von <code><body></code>
<code></code>	← Child von <code><div></code> , Parent von <code></code> , Sibling von <code><p></code>
<code>Punkt 1</code>	← Child von <code></code>
<code>Punkt 2</code>	
<code></code>	
<code><p>Etwas Text</p></code>	← Second Child von <code><div></code> , Sibling (Geschwister) von <code></code>
<code></div></code>	
<code></body></code>	



Der `<div>` Tag ist ein Container Element! (engl. division ⇔ Bereich).

Besondere Zeichen

Wenn man im Text Zeichen verwenden möchte, die in HTML eine Bedeutung haben (`<`, `>`, `&`, `"`), dann verwendet man HTML-eigene Zeichen. Beispiele dafür sind:



`<`öffnende Spitze Klammer (lower than)<
`>`schließende Spitze Klammer (greater than)>
`&`Kaufmännisches Und (Ampersand)&
`"`Anführungszeichen oben"



Die Zeichenfolge ` ` erzeugt ein geschütztes Leerzeichen (nbsp = non-breaking space = Leerzeichen ohne Umbruch).
` ` sollte man immer bei einem Element ohne Inhalt einfügen!
 Statt `<p>` `</p>` schreibt man `<p> </p>`



Ein umfangreiche Zeichenreferenz findet man bei [selfhtml.org](http://wiki.selfhtml.org/wiki/Referenz:HTML/Zeichenreferenz)
<http://wiki.selfhtml.org/wiki/Referenz:HTML/Zeichenreferenz>

HTML Kommentare

Der Browser ignoriert Kommentare und stellt diese auch nicht dar. Sie dienen dem Autor um innerhalb des Codes Anmerkungen zu platzieren oder Codeteile auszukommentieren. Eingeleitet wird ein Kommentar mit `<!--` dann folgt der Kommentartext. Abgeschlossen wird es mit `-->`

```
<!-- einzeliger Kommentar -->

<p>viel Text</p>

<!-- und das ist ein mehrzeiliger Kommentar zu dem Text mit <p>...</p>
  Letzte Zeile des Kommentars -->
```

Lernhandout 3.1 Tabellen

Referenzcode: HCL031

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... kennen den Aufbau einer Tabelle (Zeilen, Spalten, Zellen).	K	I
... kennen den Aufbau einer HTML-Tabelle.	K	II
... scripten HTML Tabellen.	I	II
... nutzen Tabellenköpfe zur besseren Gliederung.	I	II
... verbinden Spalten zu einer Tabellenzelle (colspan).	I	II
... verbinden Zeilen zu einer Tabellenzeile (rowspan).	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- Das `border` Attribut wird benutzt, um die Tabelle sichtbar zu machen. Es ist in HTML5 obsolet – sollte aber problemlos von den meisten Browsern dargestellt werden.

Man verwendet zur übersichtlichen Darstellung von Daten Tabellen. Diese sind in Zeilen (waagrecht) und Spalten (senkrecht) gegliedert. Am Schnittpunkt einer Zeile und einer Spalte ist eine Zelle. HTML-Tabellen werden nur verwendet, um tabellarische Daten darzustellen, nicht, um das Layout einer Website zu gestalten. Früher wurden Tabellen dazu eingesetzt, komplette Layouts umzusetzen. Das widerspricht allerdings letztlich dem Grundgedanken eines modernen Webdesigns.

```
<table>
  <thead>
    <tr> <th> </th> <th> </th> <th> </th> </tr>
  </thead>
  <tbody>
    <tr> <td> </td> <td> </td> <td> </td> </tr>
    <tr> <td> </td> <td> </td> <td> </td> </tr>
  </tbody>
</table>
```



`<table>` leitet eine Tabelle ein

`<thead>` Der Tabellenkopf ist für eine bessere Gliederung angedacht. Der Tag wird für die Beschriftung einer Tabelle verwendet und ist für die Darstellung nicht zwingend notwendig.

`<tbody>` Der Tabellenkörper ist ebenfalls für eine bessere Gliederung der Inhalte und kann genauso wie der `<thead>` weggelassen werden.

`<tr>` leitet eine Tabellenzeile ein (tr = table row = Tabellenzeile)

`<th>` eine Zelle im Tabellenkopf (th = table header)

`<td>` eine normale Datenzelle (td = table data)



Damit unsere Tabellen sichtbar werden, schreiben wir in den `<table>` Tag das Attribut `border="2"`. **Achtung: Diese Vorgehensweise ist in HTML5 obsolet da Rahmen mit CSS definiert werden müssen!**

Die Zellen lassen sich selbstverständlich auch verbinden. Dies wird mit den Attributen `colspan` bzw. `rowspan` im `<td>` Tag umgesetzt.

`<td colspan="3">` `colspan` (Spalte überspannen) erlaubt eine Tabellenzelle nach rechts über mehrere Spalten auszudehnen (hier über 3 Zellen)

`<td rowspan="2">` `rowspan` (Zeile überspannen) verbindet Tabellenzellen über mehrere Zeilen (hier über 2 Zeilen)

<td> </td>	<td> </td>	<td> </td>	<td> </td>	<td> </td>
<td> </td>	<td colspan="3"> </td>			<td> </td>
<td rowspan="2"> </td>	<td> </td>	<td> </td>	<td> </td>	<td> </td>
	<td> </td>	<td> </td>	<td> </td>	<td> </td>

Übungsblatt 1.1 HTML Grundgerüst

Referenzcode: HCU011

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... erstellen einen Stundenplan mit HTML.	I	II
... verwenden das colspan Attribut.	I	II
Übung B		
... nutzen Tabellen zur Gegenüberstellung von Informationen.	I	II
... binden Bilder in eine Tabelle ein.	I	II
... nutzen das Internet zur Erhebung von Informationen.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- 📄 Anmerkung zu Übung A Stundenplan: Es ist ein sehr umfangreiches Übungsbeispiel. Natürlich ist dieses Beispiel eines, welches die Eigeninitiative des|der Schüler|in wecken sollte. So gedacht, könnte man einen beliebigen Wochenplan (z. B: Reservierung eines Tennisplatzes udgl.) als Übungsbeispiel anwenden. Der Stundenplan jedoch wird noch in späteren Übungsbeispielen verwendet. Man kann das Beispiel bis zur 3 Schulstunde begrenzen (Hauptsache das colspan kommt zur Anwendung).



Übung A: Stundenplan

- ☐ Erstelle mit dem `<table>` Element deinen Stundenplan.
- ☐ Im Tabellenkopf sind folgende Beschriftungen anzuführen:

Stunde	von	bis	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag
--------	-----	-----	--------	----------	----------	------------	---------	---------
- ☐ Ändere die Unterrichtszeiten (wenn nötig) und verbinde die Zellen für die Pausen mit `colspan`
- ☐ Trage deine Unterrichtsfächer in die freien Zellen ein.

Stunde	von	bis	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag
1.	07 ⁴⁵	08 ³⁵						
2.	08 ³⁵	09 ²⁵						
	09 ²⁵	09 ⁴⁰	Pause (15 Minuten)					
3.	09 ⁴⁰	10 ³⁰						
4.	10 ³⁰	11 ²⁰						
	11 ²⁰	11 ³⁰	Pause (10 Minuten)					
5.	11 ³⁰	12 ²⁰						
6.	12 ²⁰	13 ¹⁰						
	13 ¹⁰	13 ¹⁵	Pause (5 Minuten)					
7.	13 ¹⁵	14 ⁰⁵						
8.	14 ⁰⁵	14 ⁵⁵						
	14 ⁵⁵	15 ⁰⁵	Pause (10 Minuten)					
9.	15 ⁰⁵	15 ⁵⁵						
10.	15 ⁵⁵	16 ⁴⁵						
11.	16 ⁴⁵	17 ³⁵						



Übung B: Ländervergleich

- ☐ Erstelle mit dem `<table>` Element einen Ländervergleich (Schweiz, Österreich, Deutschland) wie unten dargestellt.
- ☐ Durch einen Klick auf die Landesflagge soll sich die jeweilige Wikipedia-Seite zum passenden Land öffnen.
- ☐ Füge noch ein weiteres Land deiner Wahl dazu!

			
Land	Schweiz	Österreich	Deutschland
Amtssprache	Deutsch	Deutsch	Deutsch
Hauptstadt	Bern	Wien	Berlin
Fläche	41 285 km ²	83 878 km ²	357 385 km ²
Einwohner	8,4 Mio	8,7 Mio	82,5 Mio
Staatsform	föderale Republik	Bundesrepublik	Bundesrepublik
Währung	Schweizer Franken	Euro	Euro
Internet-TLD	.ch, .swiss	.at	.de
Telefon-Vorwahl	+41	+43	+49

Lernhandout 3.2 HTML validieren

Referenzcode: HCL032

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... kennen Unterschiede zwischen den Browsern.	I	I
... kennen das W3C Konsortium und seine Bedeutung für das Web.	I	I
... validieren (überprüfen) ihren HTML Code.	I	II
... kennen weitere Dienste des w3c (z. B. Link Checker).	I	I
... haben ein gesteigertes Bewusstsein für Genauigkeit.	A	II
... erstellen Screenshots.	I	II
... verstehen die Fehlermeldungen des Validierungsergebnisses.	I	III
... korrigieren Fehler des HTML Codes.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- 📄 Hier ist ein didaktisch interessantes Feld beheimatet. Durch die Validierung bekommen die Schüler|innen ein sofortiges Feedback auf ihre Schreibarbeit mit einer Erklärung zum Fehler bzw. der Warnung. Nur so kann das Lernkonzept von "Try-and-Fail" umgesetzt werden, da die meisten Browser fehlerhaften Code parsen und darstellen (z. B: fehlenden End-Tag). Dadurch wird der Fokus vom Lernprodukt auf den Lernprozess verlegt.

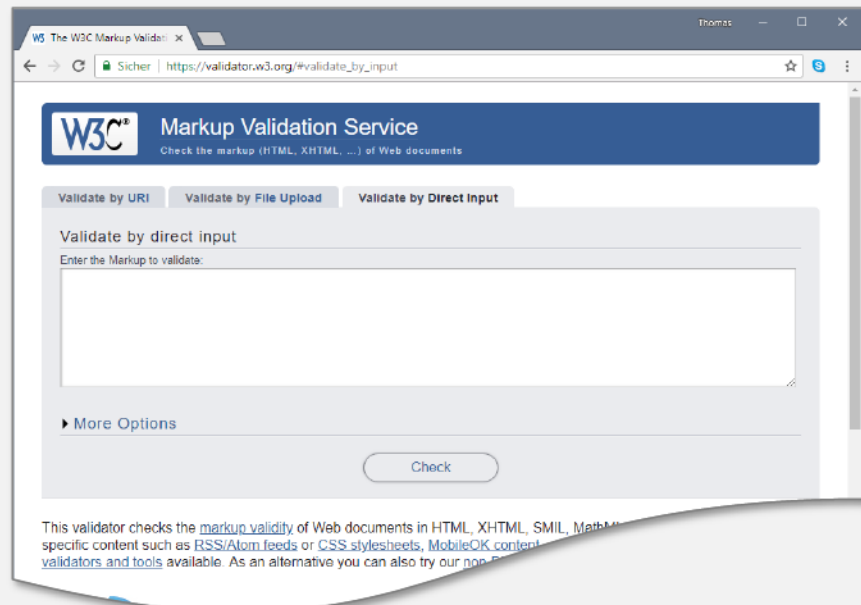
HTML Code wird in der Regel von einem Browser interpretiert. Jeder Browser hat seine Eigenheiten – Firefox ist nicht Chrome – Chrome ist nicht Edge usw. Manche Elemente oder Attribute gibt es nur für bestimmte Browser. Wir wollen aber für alle schreiben. Wie können wir nun sicher sein, dass unser Code richtig ist? Dafür gibt es das W3C. Das W3C Konsortium entwickelt Standards und Richtlinien für das Web. Die meisten Browser-Entwickler halten sich an diese Richtlinien.

Wie können wir nun aber sicher sein, dass unser Code richtig ist?

Dafür hat das W3C auf seiner Website ein **Markup Validation Service** (Prüfungsdienst, Validator) eingerichtet.

Auf www.w3.org findet man auch weitere Prüfdienste wie z. B. für CSS oder einen Link Checker.

Der HTML-Validator bietet drei Möglichkeiten an um den HTML Code zu überprüfen.



<https://validator.w3.org>



Validate by URI Man gibt die Adresse zu einem Online Dokument an.

Validate by File Upload Hier ladet man das HTML Dokument hoch.

Validate by Direct Input Man kopiert den HTML Code ins Textfeld.

Check

Nach einem Klick auf Check wird der Code überprüft und man bekommt einen umfangreichen Bericht.

- Error** End tag `<h2>` seen, but there were open elements.
From line 10, column 27, to line 10, column 31

ein Stelle `</h2>` `</bod`



Error bedeutet Fehler und muss unbedingt korrigiert werden!

Document checking completed.

Source

```
1. <!doctype html>↵
2. <html lang="de">↵
3.   <head>↵
4.     <meta charset="utf-8">↵
5.     <title>Übungen 4.3</title>↵
6.     <meta name="author" content="Thomas Maier">↵
7.     <meta name="date" content="2018-04-26">↵
8.   </head>↵
9.   <body>↵
10.    <h1>Führerschein Stelle</h2>↵
11.  </body>↵
12. </html>
```



Übung A: HTML Code validieren

- ☐ Überprüfe den HTML Code von der **Übung 2.2 A: Ein Inhaltsverzeichnis**.
- ☐ Erstelle einen Screenshot vom Prüfergebnis.



Screenshots erstellt man durch einen Klick auf die Druck Taste deiner Tastatur. Hilfreiche Tools um Screenshots zu erstellen sind das **Snipping Tool** oder **Sketch**.

Lernhandout 4.1 style, span, color

Referenzcode: HCL041

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... arbeiten zum ersten Mal mit CSS.	I	II
... nutzen das style Attribut für Inline-Styles.	I	II
... kennen den Aufbau von CSS Eigenschaften.	K	I
... verändern Schriftfarben mit CSS.	I	II
... verändern die Hintergrundfarbe eines HTML Elements mit CSS.	I	II
... arbeiten mit dem Tag.	I	II
... kennen die Möglichkeiten zur Farbuweisung. (Farbname, RGB, HEX)	I	II
... generieren eigene Farben durch RGB bzw. HEX.	I	IV
... verstehen weitere Farbmodelle (z. B: CMYK für den Druck)	K	I

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

CSS Styles

Bislang hatten unsere HTML Seiten nur das Standard Erscheinungsbild des Browsers. Um nun einem Element ein besonderes Design zu verleihen, verwendet man das *style* Attribut. Das *style* Attribut gilt für nahezu alle HTML Tags und hat folgende Schreibweise:

HTML	<code><tagname style="property:value;"></code>
CSS	<code>color: [Wert];</code> ← Schriftfarbe (siehe Farben unten)
<code><p style="color: red;">Die Schrift ist jetzt rot</p></code>	



property ist der CSS-Befehl (Eigenschaft), *value* ist der Wert. Wichtig ist das Semikolon (;) am Ende jeder CSS Anweisung. Im folgenden Beispiel wird die Schriftfarbe des Textes eines `<p>` Tags auf die Schriftfarbe rot gesetzt.

Man kann auch mehrere CSS Anweisungen (Stile) mit dem *style* Attribut zuweisen.

CSS	<code>background-color: [Wert];</code> ← Hintergrundfarbe (siehe Farben unten)
<code><p style="color: white; background-color: blue;"> Der Schnee ist weiß und der Himmel ist blau.</p></code>	

HTML	<code> ... </code>
Wenn man nur ein Wort oder eine Textstelle auszeichnen möchte, benutzt man das <code></code> Element (<i>span</i> = engl. überspannen). Der <code></code> Tag alleine bewirkt nichts – erst in Kombination mit CSS erfüllt er eine Aufgabe!	
<code><h1>Die Lehre über Energie und Wärme</h1></code>	

Farben

CSS	<code>color: [Wert];</code>
 Um den WERT (value) für Farben festzulegen, gibt es 3 wichtige Möglichkeiten:	
1. Farbnamen	Die Namen werden in Englisch ausgeschrieben <code>color: DeepSkyBlue;</code>
2. RGB	RGB (Rot, Grün, Blau) ist das Standard Farbmodell für Bildschirme. Die einzelnen Werte reichen von 0 bis 255. <code>color: rgb(0, 191, 255);</code>
3. HEX	Hier werden die RGB Werte Hexadezimal dargestellt. <code>color: #00bfff;</code>
 Weitere Farbmodelle sind HSL, HWB und CMYK. CMYK (Cyan, Magenta, Yellow und Schwarz) ist ein Farbmodell für den Druck.	

Lernhandout 4.2 Rahmen

Referenzcode: HCL042

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... definieren Rahmen durch Dicke, Typ und Farbe.	I	II
... bestimmen einzelne Rahmenbereiche (z. B: oben durch border-top).	I	II
... runden Rahmenecken mit border-radius ab.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- 📄 Die CSS Eigenschaft `border` ist eine 'shorthand-property' und verbindet `border-width`, `border-style`, `border-color` zu einer Zuweisung.

Mit der *border* Eigenschaft (CSS) kann man den Rahmen um ein Element bestimmen.
border hat drei Werte: **Rahmendicke**, **Rahmentyp** und **Rahmenfarbe**!
 Zwischen den Werten wird ein Leerzeichen gesetzt. Am Ende ein Semikolon (;).

CSS `border: [dicke] [typ] [farbe];`



[dicke] Die Rahmendicke kann in Pixel angegeben werden (z. B: 5px) bzw. *thin* (dünn) *medium* (mittel) oder *thick* (dick). Die Rahmendicke alleine reicht noch nicht für eine Darstellung – es muss der Rahmentyp angegeben werden!

[typ] Folgende Rahmentypen kennt CSS:

solid.....fester Rahmen, ohne besondere Erscheinung

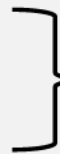
dotted.....gepunkteter Rahmen

groove

ridge

inset

outset



simuliert einen 3D Effekt

double.....doppelter Rahmen

dashed.....gestrichelter Rahmen

Das Beispiel oben erzeugt einen doppelten Rahmen in der Stärke von 8 Pixel um das `<h1>` Element.

```
<h1 style="border: 8px double;">Kapitel 16 ...</h1>
```



[farbe] Nun kann man die Eigenschaft noch um einen Farbwert (siehe 4.1) ergänzen. (#00F ist der Hex-Wert für Blau).

```
<h1 style="border: 10px dotted #00F;">Deutsche Automarken</h1>
```



Man kann auch einzelne Rahmen bestimmen:

border-top für den oberen Rahmen. *border-bottom* für den unteren Rahmen.
border-left für den linken und *border-right* für den rechten Rahmen.

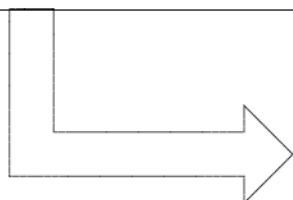
```
<h1 style="border-bottom: 5px solid red;">Unser Jahrbuch</h1>
```

CSS `border-radius: [li-oben] [re-oben] [re-unten] [li-unten];`



Der Wert wird in Pixel angegeben. Gibt man nur einen Wert an, dann werden alle vier Ecken gleichmäßig abgerundet. Die Ecken werden im Uhrzeigersinn definiert.

```
<h1 style="border: 10px solid rgb(255,126,0);  
border-radius: 25px 0px 25px 0px;  
background-color: blue;  
color: white;">... in einem fernem Land <br>  
... vor gar nicht allzulanger Zeit! </h1>
```



Lernhandout 4.3 Höhe und Breite

Referenzcode: HCL043

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... nutzen die CSS Eigenschaften height (Höhe) und width (Breite).	I	II
... haben ein Wissen über absolute Längenmaße.	K	I
... haben ein Wissen über relative Längenmaße.	K	I
... unterscheiden zwischen absoluten und relativen Längenmaßen.	K	II
... arbeiten mit den passenden Längenmaßen nach pragmatischen Gesichtspunkten.	K	III
... begrenzen Höhe und Breite durch minimal oder maximal Werte (min-height, max-height).	I	II
... steigern ihr Layout-Bewusstsein (z. B. Goldener Schnitt, Proportionslehre).	A	I

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Die CSS Eigenschaften `height` (Höhe) und `width` (Breite) bestimmen die Größenangaben eines HTML Elements.

CSS `height: [Wert]; width: [Wert];`



[Wert] Für die Größenangaben gibt es in CSS eine breite Palette an Möglichkeiten. Wir unterscheiden zwischen absoluten und relativen Längenmaßen.

Absolute Längenmaße sind fixe Größe und ändern sich in der Regel nicht!

px	Pixel	Bildpunkte des Screens (Bildschirm, Smartphone udgl.)
cm	Zentimeter	Entspricht ca. 37,8 Pixel
mm	Millimeter	Entspricht ca. 3,78 Pixel, bzw. 0,1 cm
in	Zoll	Ein Zoll sind 2,54 cm oder 96 Pixel
pt	Punkt	Werden für Schriftgrößen verwendet. 1 pt entspricht ca. 1,33 Pixel

```
<p style="background-color: blue; width: 4cm;">&nbsp;&nbsp;&nbsp;</p>
```

Relative Längenmaße beziehen sich auf die Größe eines Elements (meist dem Elternelement). Wird die Größe verändert – z. B. Browserfenster verkleinern/vergrößern, dann verändert sich auch die Größen des Kind Elements – und zwar relativ.

%	Prozent %	z. B. <code>width: 50%;</code>
em	em	Vertikale Größe einer Schrift. Praktisch für Schriftarten.
rem	Wurzel-Em	Relativ zum Wurzelement (<code><html></code>).
vw	Viewport-Breite	entspricht dem 100. Teil des Anzeigebereichs. <code>width: 50vw;</code> ist halb so groß, wie der Screen (bzw. die Browserbreite).
vh	Viewport-Height	entspricht dem 100. Teil der Höhe des Anzeigebereichs. <code>height: 25vh;</code> hat ¼ der Höhe des Screens.

```
<div style="border: 5px solid; height: 50vh;">&nbsp;&nbsp;&nbsp;</div>
```

Das Beispiel oben zeigt das Container-Element `<div>` mit der halben Höhe des Browser-Fensters an. Verändert sich die Browsergröße (nicht jeder Browser wird in Vollbild gestartet), dann verändert sich auch die Höhe des `<div>` Elements.

Relative Größen haben das Problem, dass sie entweder zu groß oder zu klein sind wenn sich die Bezugsgröße verändert. Dafür gibt es die Eigenschaften:



`max-width` und `max-height` Welche die maximale Breite oder Höhe beschreiben. z. B. eine relative Breite von 50% endet bei 300 Pixel, wenn man `max-width: 300px` festgelegt hat.

`min-width` und `min-height` geben jene Größen an, die auf jeden Fall dargestellt werden sollen. Dieses Minimum darf nicht unterschritten werden!

```
<div style="background-color: green;
width: 50%;
min-width: 200px;">&nbsp;&nbsp;&nbsp;</div>
```

Übungsblatt 4.3 Farben, Rahmen und Größenangaben

Referenzcode: HCU043

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... reproduzieren eine graphische Darstellung laut Vorlage.	I	II
... verschachteln Elemente zur Lösung der Aufgabe.	I	III
Übung B		
... erstellen eine Tabelle mit farblichen Hintergründen.	I	II
... finden RGB und HEX Werte selbstständig heraus.	I	II
Übung C		
... erstellen einfache HTML Buttons.	I	II
... steigern ihr Kreativbewusstsein durch freie Arbeit.	A	II
... verwenden Rahmen und Farben.	I	II
Übung D		
... erstellen eine Verkehrsampel ausschließlich mit HTML und CSS.	I	II
... verwenden relative Größenangaben.	I	II
... verstehen, dass man mit Eckenabrundungen einen Kreis erzeugen kann.	K	III

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen



Übung A: Loveparade

- ☐ Erstelle nur mit HTML und CSS (style="" Attribut) einen Loveparade-Schriftzug wie unten dargestellt.
- ☐ Bilddateien sind **nicht** erwünscht!
- ☐ Verwende nur ein einziges <h1> Element.
- ☐ Jeder Buchstabe hat eine eigene Hintergrundfarbe.
- ☐ TIPP: Verwende Tags und die Zeichenreferenz!



Übung B: Farbtabelle

- ☐ Erstelle nur mit HTML und CSS eine Farbtabelle.
- ☐ Ermittle die RGB und HEX Werte der fehlenden Farben und trage Sie in die Tabelle ein! (Türkis, Blaugrün, Himmelblau, Magenta)

Farbname	R	G	B	HEX
Gold	255	215	0	#FFD700
Gelbgrün	154	205	50	#9aCd32
Türkis				
Blaugrün				
Himmelblau				
Magenta				



Übung C: Einfache Buttons

- ☐ Designe drei unterschiedliche Button!
- ☐ FREIES Design (alles ist erlaubt, solange es nur in HTML und CSS ist, keine Bilder!!!)
- ☐ Die Buttons haben noch keine Funktion – noch nicht!!!

Bestellung absenden

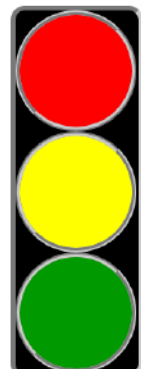
Bestellung abbrechen

Bestellung neu erstellen



Übung D: Eine Ampel

- ☐ Erstelle nur mit HTML und CSS eine Verkehrsampel.
- ☐ Die Größe der Ampel wird durch die Breite des Browsers bestimmt.
(z. B. wird der Browser in der Breite größer, dann soll die Ampel auch größer werden – sie skaliert).
- ☐ TIPP: Verwende `border-radius` und relative Größenangaben!!!



Lernhandout 5.1 CSS im Head

Referenzcode: HCL051

Feinziele Die Schüler innen ...	Zielart	Taxonomie
... verlagern die CSS Anweisungen in den <head> des HTML Codes.	I	II
... wissen, was ein Selektor ist.	I	I
... kennen Typselektoren um HTML Elemente anzusprechen.	I	I
... verwenden Typselektoren mit der korrekten Syntax.	I	II
... kennen ID-Selektieren um ein bestimmtes Elemente anzusprechen.	I	I
... verwenden ID-Selektoren mit der korrekten Syntax.	I	II
... kennen Klassenselektoren um mehrere gleiche Elemente anzusprechen.	I	I
... verwenden Klassenselektoren mit der korrekten Syntax.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Bisher haben wir unsere CSS Anweisungen direkt in das HTML Element geschrieben. Praktischer ist es jedoch, die CSS Anweisungen innerhalb des `<head>` Tags zu schreiben. Dafür öffnet man mit `<style>` und schließt dann mit `</style>`. Für die Zuweisung der CSS Stile gibt es drei Möglichkeiten (drei Selektoren: Tag, ID und Klasse).

```
<head>
...
<style>
  h1 {border-bottom: thin solid black;}
  #hauptinhalt {background-color:#CFD5EB; color:#00C;}
  .wichtig {color:red;}
</style>
</head>
```

1. Ein HTML Element überschreiben (Typselektor)

Man definiert für einen Tag ein neues Aussehen. Jedesmal wenn dieser Tag dann im HTML-Dokument verwendet wird, erhält er das Aussehen aus dem `<style>` im `<head>`.

Schreibweise CSS

```
TAG {CSS-Anweisung;}
```

```
CSS:
h2 {color:red;}

HTML:
<h2>Empfehlungen</h2>
```

2. ID festlegen (ID-Selektor)

Eine ID (Identifikation) gilt nur für ein einziges HTML Element. Im gesamten Code darf die ID nur einmal vorkommen. Im CSS schreibt man eine Raute (#) vor dem Namen. Ins HTML Element schreibt man das Attribut `id=" "`

Schreibweise CSS

```
#ID {CSS-Anweisung;}
```

```
CSS:
#haupt {background-color:blue;}

HTML:
<p id="haupt">Empfehlungen</h2>
```

3. Eine Klasse definieren (Klassenselektor)

Mit einer Klasse kann man beliebig viele Elemente ansprechen. Eine Klasse kann auch auf unterschiedliche Tags angewandt werden. Im CSS schreibt man einen Punkt vor den Klassennamen. Ins HTML Element schreibt man das Attribut `class=" "`

Schreibweise CSS

```
.CLASS {CSS-Anweisung;}
```

```
CSS:
.srl {color:red;}

HTML:
<h1 class="srl">Vorlagen</h1>
<span class="srl">WICHTIG</span>
```



Achte bei den Namen für die Selektoren auf die Groß- und Kleinschreibung.

Lernhandout 5.2 Schriftarten

Referenzcode: HCL052

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... verstehen wie Browser auf Schriftarten zugreifen.	I	I
... wissen was Schriftfamilien und generische Schriftarten sind.	I	I
... verwenden die korrekte CSS Syntax um eine Schriftfamilie zu definieren.	I	II
... kennen die Kategorien von Schriften.	I	I
... steigern ihre Kreativität durch Kenntnisse der Typographie.	A	II
... binden eigene Schriftarten ein.	I	II
... beachten die Urheberrechte von Schriftarten.	K	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- 📄 Urheberrechte sind ein wichtiges Thema und sollten mehrmals zur Sprache gebracht werden. Auch die CC (Common Creatives) sollte thematisiert werden.

Um eine Schriftart für ein Element zu bestimmen, verwenden wir `font-family`. Der Browser greift auf die installierten Schriftarten des Empfänger-System zu. Da es sein kann, dass der Endbenutzer nicht jede Schriftart installiert hat, geben wir mehrere Schriftarten als Wert mit. Zusätzlich noch eine Generische Schriftart (allgemeine Schriftfamilie). Die Schriftnamen werden durch einen Beistrich getrennt. Wie bei allen anderen CSS Anweisungen endet `font-family` ebenfalls mit einem Semikolon (;). Besteht eine Schriftart aus mehreren Wörtern, dann wird sie unter Anführungszeichen (") geschrieben.

CSS `font-family: [Schriftname] ["Schrift 2"] [Generische Schrift];`



[Generische Schrift] Folgende generische Schriftfamilien können ausgewählt werden:

<code>serif</code>	eine Schriftart mit Serifen (z. B. Times New Roman, Palatino)
<code>sans-serif</code>	ohne Serifen (z. B. Univers , Calibri)
<code>cursive</code>	Schreibschriftarten (z. B. <i>Mistral</i> , <i>Vivaldi</i>)
<code>fantasy</code>	ungewöhnliche Schriften (z. B. IMPACT LABEL , ROSEWOOD)
<code>monospace</code>	alle Zeichen haben die gleiche Breite (z. B. Courier, Consolas)

```
h1 {font-family:Tahoma, Geneva, sans-serif;}
```

```
<h1>Automobilmuseum</h1>
```

Will man nun eine eigene Schriftart verwenden, dann benutzen wir die CSS Anweisung `@font-face`. Innerhalb der `@font-face` Anweisung, wird der Name der Schriftfamilie, eine gültige URL zur Schriftart (wo ist diese abgespeichert?) und das Format der Schriftart.

```
@font-face {font-family: 'neue-schrift';
            src: url('pfad/zu/neueschrift.ttf') format('truetype'); }
```

Der Name der Schriftfamilie muss nicht immer `'neue-schrift'` lauten. Es kann ein beliebiger Name gewählt werden – jedoch bitte kein Name einer schon bestehenden anderen Schriftart. Danach kann die Schriftart mit `font-family` zugewiesen werden!

```
<p style="font-family: neue-schrift, Arial, sans-serif">
  ... Symbole helfen das ...</p>
```



Achte auf die Dateigrößen einer Schriftart. Schriftarten von mehr als 300 KiB verlangsamen die Ladegeschwindigkeit. Nach der erstmaligen Abholung wird die Schriftart im Browsercache gespeichert.

Urheberrechte beachten



Schriftarten sind durch Urheberrechte geschützt. Ohne eine Genehmigung (erworben/gekauft, CC, gratis Schriftarten) kann die Nutzung im Internet teuer werden. Manche Schriftarten sind technisch geschützt und werden im Browser nicht angezeigt.



Google bietet eine Vielzahl von freien und offenen Schriftarten auf <https://fonts.google.com> an. Diese Schriftarten lassen sich praktisch und einfach in eine Webseite einbinden!

Lernhandout 5.3 Schriftgrößen

Referenzcode: HCL053

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... verändern die Schriftgrößen mit CSS.	I	II
... unterscheiden zwischen absoluten und relativen Schlüsselwörtern.	K	II
... verwenden auch die allgemeinen CSS-Größenangaben.	I	II
... entscheiden welche Angaben für Druck bzw. für Bildschirmdarstellung passend sind.	I	III
... verändern die Strichstärke eines Textes.	I	II
... verwenden CSS um einen Text in Kapitälchen darzustellen.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- Im Bereich `font-variant` wird als Beispieltext: **Was it the cat i saw** verwendet. Dieses schöne Palindrom ist aus Alice im Wunderland. Es ist zwar äußerst ästhetisch, hat aber wenige Großbuchstaben um die Kapitälchen besser zu illustrieren.

CSS `font-size: [Wert];`

Mit `font-size` bestimmt man die Darstellungsgröße der Schrift. Als Werte können alle Größe (siehe 4.3 Höhe und Breite) angewendet werden. Zusätzlich gibt es noch Schlüsselwörter:

Schlüsselwörter mit absoluten Angaben

<code>xx-small</code>	<i>winzig</i>	<code>large</code>	<i>groß</i>
<code>x-small</code>	<i>sehr klein</i>	<code>x-large</code>	<i>sehr groß</i>
<code>small</code>	<i>klein</i>	<code>xx-large</code>	<i>riesig</i>
<code>medium</code>	<i>mittel</i>		

Schlüsselwörter mit relativen Angaben

<code>smaller</code>	<i>Kleiner als das Elternelement</i>
<code>larger</code>	<i>Größer als das Elternelement</i>
<code>inherit</code>	<i>Die gleiche Größe wie das Elternelement.</i>



Absolute Angaben (z.B. pt, px) eignen sich für den Druck und relative Angaben für Bildschirmdarstellungen. Die Schriftgröße eines Elements wird in der Regel automatisch an das Kindelement vererbt.

```
h2 {font-size: 45px;}
#haupttext {font-size: medium;}
.tipp {font-size: larger;}
span {font-size: 1.2em;}
```

CSS `font-weight: [Wert];`

Mit `font-weight` wird die Strichstärke (Dicke) des Textes festgelegt. Folgende Werte sind möglich:

<code>lighter</code>	<i>dünner als im Elternelement</i>
<code>normal</code>	<i>normale Strichstärke</i>
<code>bold</code>	<i>fett</i>
<code>bolder</code>	<i>fetter als im Elternelement</i>
<code>100, 200, 300 ... 900,</code>	<i>extra-dünn (100) bis extrafett (900)</i>
<code>inherit</code>	<i>Strichstärke des Elternelements</i>

```
.fetter {font-weight: bolder;}
#impressum {font-weight: 400;}
```

CSS `font-variant: [Wert];`

Mit `font-variant: small-caps;` kann man einen Text mit Kapitälchen darstellen. Kapitälchen sind Großbuchstaben in der Höhe von Kleinbuchstaben.

Beispiel ohne Kapitälchen: Was it the cat i saw?

Beispiel mit Kapitälchen: WAS IT THE CAT I SAW?

```
<p>Was it the cat i saw?</p>
<p style="font-variant: small-caps;">Was it the Cat i saw?</p>
```

Lernhandout 5.4 Text Dekoration

Referenzcode: HCL054

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... über-, durch- oder unterstreichen Text mit CSS.	I	II
... kennen die Attribute der CSS Eigenschaft text-decoration.	I	I
... verändern das Aussehen von Hyperlinks.	I	II
... achten bei Unterstreichungen auf die Unterlänge von Buchstaben.	K	II
... definieren Textschatten mit CSS.	I	II
... entwickeln ein Designverständnis für die Verwendung von Textdekorationen. (z. B. sparsamer Umgang mit Schatten).	KA	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Mit der `text-decoration` Eigenschaft kann man einen Text über-, unter- oder durchstreichen.

```
<h1 style="text-decoration: underline;">Unsere Fahrpläne</h1>
```

Zusätzlich können Stiel und Farbe der Linien definiert werden! Für Color-Angaben gelten die allgemeinen Regeln (siehe 4.1 `style`, `span`, `color`).

CSS `text-decoration: [line] [style] [color];`



[line]

underline.....*unterstrichen*
 overline.....*überstrichen*
 line-through.....*durchgestrichen*
 none.....*keine Textdekoration*



[style]

solid.....*durchgezogene Linie* double.....*doppelte Linie*
 dotted.....*gepunktete Linie* dashed.....*gestrichelte Linie*
 wavy.....*Wellenlinie*

```
.falsch {text-decoration: underline wavy red;}
```

```
<p>Standard wird oft mit einem t geschrieben  

  <span class="falsch">(Standart)</span></p>
```

Hyperlinks (`a href`) werden vom Browser meist blau mit einer durchgezogenen Linie dargestellt. Will man die Unterstreichung von den Links entfernen, dann kann das mit `text-decoration: none;` umgesetzt werden.

```
a {text-decoration:none;}
```



Design Tipp: Achte darauf, ob die Unterstreichung die Unterlänge des Textes nicht übermalt. Buchstaben mit Unterlänge sind: g, j, y usw.

Textschatten

Mit der `text-shadow` Eigenschaft fügt man dem Text einen Schatten hinzu.

CSS `text-shadow: [H] [V] [blur] [color];`



[H] Die Horizontale Position des Schattens. (Negative Werte sind erlaubt)
 [V] Die vertikale Position des Schattens. . (Negative Werte sind erlaubt)
 [blur] Radius der Unschärfe (Weichzeicheneffekt).
 [color] Farbangabe

```
h1 {text-shadow: 5px 5px 10px blue;}  

h2 {color: white; text-shadow: 2px 2px 4px #000000;}
```

```
<h1>Moderne Medizin</h1>  

<h2>Die chinesische Medizin im Überblick</h2>
```


Übungsblatt 5.4 Schriften und Textdekoration

Referenzcode: HCU054

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagment).	S	II
... reorganisieren die CSS Anweisungen in den <head>.	I	II
Übung B		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagment).	S	II
... ändern die Schriftart von Buttons.	I	II
... suchen im Internet nach freien Schriftarten und beachten dabei das Urheberrecht.	IK	III
... binden Schriftarten mit CSS ein.	I	II
Übung C		
... recherchieren die Nutzung von Google-Fonts und stellen dabei Überlegungen zur praktischen Umsetzung an.	I	III
... dokumentieren das Rechercheergebnis.	I	II
Übung D		
... wandeln einen Plain-Text in ein HTML Dokument.	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen



Übung A: Einfache Buttons im Head

- ☐ Öffne deine Arbeit/Lösung von **4.3 C: Einfache Buttons**
- ☐ Schreibe die Arbeit so um, dass alle CSS Anweisungen im Head sind!
- ☐ Solltest du 4.3.3. noch nicht gelöst haben, dann erstelle drei Buttons mit CSS Anweisungen im Head.



Übung B: Buttons im Head

- ☐ Öffne deine Arbeit/Lösung von **5.4 A: Einfach Buttons im Head.** (siehe oben)
- ☐ Erstelle einen weiteren Button – dieser soll eine außergewöhnliche Schriftart haben. (also keine Standard Schriftart).
 - o Suche dafür im Internet nach einer freien Schriftart die du downloaden kannst und die dir gefällt. Beachte die Urheberrechte!
 - o Binde die neue Schriftart mit CSS ein. Tipp: @font-face
 - o Der neue Button soll als Klasse verfügbar sein und vier mal dargestellt werden. Tipp: class=" "



Übung C: Google Fonts

- ☐ Finde heraus, wie man die Schriftarten von font.google.com nutzen kann!
- ☐ Dokumentiere es schriftlich. (z. B. mit Word, PowerPoint, HTML ...)



Übung D: Österreich

- ☐ Öffne die Austria.txt
- ☐ Kopiere den Inhalt der Austria.txt in den <Body> eines neuen HTML Dokuments.
- ☐ Formatiere den Text bzw. das Dokument mit CSS (Freie Übung).
- ☐ Speichere die fertige Arbeit als Austria.html

ÖSTERREICH

Österreich (amtlich Republik Österreich)

Das Bundesland Wien ist zugleich Bundeshauptstadt und einwohnerstärkste Stadt des Landes. Weitere Bevölkerungszentren sind Graz, Linz, Salzburg und Innsbruck.

Österreich im Überblick

- Amtssprache: Deutsch
- Hauptstadt: Wien
- Staatsform: Bundesrepublik
- Fläche: 83.878,99 km²
- Einwohnerzahl: 8.699.730
- Währung: Euro € (EUR)

Lernhandout 6.1 Ausrichtung

Referenzcode: HCL061

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... bestimmen die Ausrichtung von Textteilen.	I	II
... verwenden die text-align Eigenschaft für Links- und Rechtsbündig, Zentriert und Blocksatz.	I	II
... erzeugen Erstzeileneinzüge mit text-indent.	I	II
... definieren Spalten für Bereiche mit viel Text.	I	II
... nutzen weitere CSS Eigenschaften in Verbindung mit Spalten. (column-gap, column-rule, column-rule).	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Natürlich muss der Text nicht immer Linksbündig sein. Mit der `text-align` Eigenschaft (CSS) kann der Text (ebenso andere inline-Element, z.B. ``) auch Zentriert, Rechtsbündig oder als Blocksatz dargestellt werden.

-  Linksbündig `left`
-  Zentriert `center`
-  Rechtsbündig `right`
-  Blocksatz `justify`

CSS `text-align: [Wert];`



Für [Wert] können folgende Werte eingesetzt werden:

- `left` Linksbündig
- `center` Zentriert
- `right` Rechtsbündig
- `justify` Blocksatz (mit `start` und `end` kann man noch das Verhalten der letzten Zeile bestimmen).

```
<h1 style="text-align:center;
border-bottom:3px solid;">Mitteilung</h1>
<p style="text-align:right; font-size:smaller">by John Doe</p>
```

Einen **Erstzeileneinzug** erzeugt man mit `text-indent`. Als Wert [value] sind positive oder negative absolute Längenangaben, oder prozentual % relativ zur Breite.

CSS `text-indent: [Wert];`

```
<p style="text-indent: 5%;" >Auf der Registerkarte ...</p>
```

Spalten

Besonders viel Text lässt sich mit Spalten übersichtlicher gestalten – besonders wenn es zu einer Darstellung der Webseite auf breiten Monitoren kommt.

CSS `columns: [count] [width];`



- [count] Wie viele Spalten sollen angezeigt werden? Eine positive Zahl oder `auto`. z. B. 2 für 2 Spalten.
- [width] Eine Längenangabe für die Mindestspaltenbreite. z. B. 6em, 500px, Zwischen den zwei Werten wird ein Leerzeichen gesetzt.

```
.absatz {columns:3 7em; text-align:justify;}
```



Weitere CSS Eigenschaften in Verbindung mit Spalten (`columns`):

- `column-gap` bestimmt den Abstand zwischen den Spalten.
- `column-rule` um eine Linie zwischen den Spalten zu bestimmen.

!!! ACHTUNG: Beide Eigenschaften sind noch nicht in allen Browsern implementiert !!!

CSS `column-rule: [width] [style] [color];`



- [width] Linienstärke, [style] alle border-style-Werte, [color] alle Farben.

```
.absatz {columns: 2; column-gap: 1.8em; column-rule: 5px dotted red;}
```

Lernhandout 6.2 Abstände

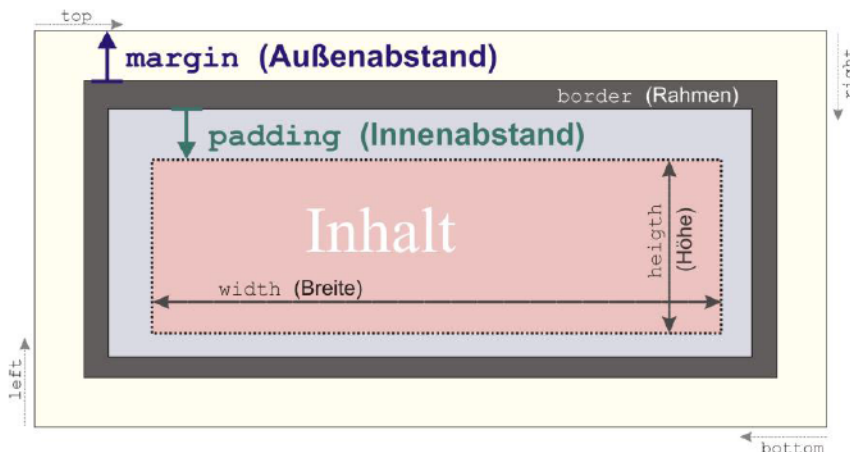
Referenzcode: HCL062

Feinziele Die Schüler innen ...	Zielart	Taxonomie
... kennen und verstehen das "klassische" Boxmodell.	I	I
... definieren den Außenabstand einer Box mit der CSS Eigenschaft margin.	I	II
... definieren den Innenabstand einer Box mit der CSS Eigenschaft padding.	I	II
... setzen die Abstände an einer bestimmten Länge oder Breite (top, right, bottom, left).	I	II
... erkennen die Auswirkungen eines Innenabstands (padding) auf die Größe der Box.	I	II
... entfernen die Abstände des <body> Elements.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Im "klassischen" Boxmodell wird eine Box (rechteckiges HTML Element z. B. `<div>` oder `<h1>`) durch seine Höhe und Breite, durch den Innen- und Außenabstand sowie durch den Rahmen bestimmt. Mit der CSS Eigenschaft `margin` bestimmt man den Außenabstand – mit `padding` den Innenabstand. Beide beziehen sich auf den Rahmen (auch wenn kein Rahmen definiert wurde!)



CSS `padding: [top] [right] [bottom] [left];`



- `[top]` Innenabstand oben. (auch `padding-top:`).
- `[right]` Innenabstand rechts (auch `padding-right:`).
- `[bottom]` Innenabstand unten (auch `padding-bottom:`).
- `[left]` Innenabstand links (auch `padding-left:`).

Als Werte sind alle numerischen Längenmaße erlaubt (z. B. cm, px, em). Negative Werte sind nicht erlaubt. Gibt man nur einen Wert an, dann gilt dieser für alle vier Seiten.

```
h1 {padding-left: 3em;}
td {padding: 10px 5px 5px 10px;}
```

Der Außenabstand (`margin`) funktioniert gleich wie `padding`. Man kann den Außenabstand auch einzeln ansprechen (`margin-top`, `margin-right` usw.). Zusätzlich erlaubt die `margin` Eigenschaft noch den Wert `auto` und negative Längenmaße.

CSS `margin: [top] [right] [bottom] [left];`

```
.wichtig {margin: 10px 4px 10px 5px; border: 2px solid;}
h2 {margin: -10px;}
```



Um ein Box-Element zentriert dazustellen muss man den `margin` Wert auf `auto` setzen und eine Breite angeben!

```
.dieBox {margin:auto; width:600px;
background:#FC0; height:3em; border:2px solid; }

<h1 class="dieBox">Wir verwenden Cookies!</h1>
<div class="dieBox" style="text-align: right;">Ist das Okey?</div>
```



Die meisten Browser stellen das `body` Element mit einem Seitenrand dar. Möchte man eine Seite bis zu den äußersten Rändern des darstellbaren Bereichs anzeigen, muss man den `margin` Wert auf 0 setzen. `body {margin: 0px;}`

Übungsblatt 6.2 Ausrichtung und Abstände

Referenzcode: HCU062

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... führen Absatz- und Zeichenformatierungen durch.	I	II
... zentrieren das Tabellenelement über die Bildschirmbreite (margin: auto;).	I	II
... definieren Abstände, Ausrichtungen und Rahmen.	I	II
... arbeiten strukturiert nach Vorgaben.	K	II
Übung B		
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
Übung C		
... arbeiten und orientieren sich in einem fremden HTML Dokument.	KI	III
... führen Absatz- und Zeichenformatierungen durch.	I	II
... definieren Abstände und Ausrichtungen.	I	II
... arbeiten strukturiert nach Vorgaben.	K	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln




Anmerkungen

- 📄 Die Startdatei Wordelemente.html muss zum Download bereitgestellt werden!



Übung A: Ländervergleich

- ☐ Öffne deine HTML-Arbeit **3.1 B: Ländervergleich** (Tabellenübung).
- ☐ Die Ländersnamen sollen hervorgehoben sein (Font-weight, Font-size).
- ☐ Die Zellen zu den Ländern sollen zentriert sein – die Beschriftungszellen (Land, Amtssprache, Hauptstadt usw.) bleiben linksbündig.
- ☐ Zwischen den Zeilen soll ein "Rahmenstrich unten" sein.
- ☐ Die Beschriftungszellen sollen einen grauen Hintergrund haben.
- ☐ Ändere die Schriftart und Schriftgröße der Tabelle (z. B. Arial, Verdana).
- ☐ Füge einen Innenabstand für Zellen hinzu.
- ☐ Zentriere die gesamte Tabelle über die Browserbreite! Tipp: `margin`
- ☐ Die Tabelle soll einen dicken groovie Rahmen haben.

			
<i>Land</i>	Schweiz	Österreich	Deutschland
<i>Amtssprache</i>	Deutsch	Deutsch	Deutsch
<i>Hauptstadt</i>	Bern	Wien	Berlin
<i>Fläche</i>	41 285 km ²	83 878 km ²	357 385 km ²
<i>Einwohner</i>	8,4 Mio	8,7 Mio	82,5 Mio
<i>Staatsform</i>	föderale Republik	Bundesrepublik	Bundesrepublik
<i>Währung</i>	Schweizer Franken	Euro	Euro
<i>Internet-TLD</i>	.ch, .swiss	.at	.de
<i>Telefon-Vorwahl</i>	+41	+43	+49



Übung B: Stundenplan

- ☐ Öffne deinen Stundenplan (siehe **3.1 A: Stundenplan**).
- ☐ Verschönere ihn mit CSS – freie Kreativübung.



Übung C: Text mit Spalten

- ☐ Öffne die Datei `wordelemente.html`
- ☐ Lege um den gesamten Text ein `<div>` Element mit der `id="Hauptdiv"`.
- ☐ Das Hauptdiv hat eine Breite von 75%, schwarzen Hintergrund und weiße Schrift. Einen Innenabstand (`padding`) von 2em.
- ☐ !!! Das Hauptdiv soll ganz **Rechts am Browserrand** ausgerichtet werden. !!!
- ☐ Der Seitenrand vom `body` soll entfernt werden.
- ☐ Alle Absätze sind linksbündig ausgerichtet und haben einen Erstzeileneinzug von 3em.
- ☐ Der zweite Absatz (Lorem ipsum dolor sit ...) soll mit drei Spalten und als Blocksatz dargestellt werden.

Lernhandout 6.3 Best Practice

Referenzcode: HCL063

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... binden externe Stylesheets in ein HTML Dokument ein.	I	II
... organisieren Dateien um Ordnung zu schaffen.	S	II
... assoziieren ein einfaches Verständnis für Corporate Design (viele HTML Seiten mit demselben Aussehen).	K	III
... unterscheiden CSS Stylesheets nach Ausgabemedien (screen, print).	K	I
... definieren für mehrere Selektoren die gleichen CSS Eigenschaften.	I	II
... weisen einem HTML Element mehrere CSS Klassen zu.	I	II
... kennen den CSS Universalselektor.	I	I

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Bisher haben wir unsere CSS Anweisungen entweder im `<head>` oder im HTML Element selbst geschrieben. Was aber, wenn man nicht nur eine Webseite hat, sondern viele? Natürlich will man, dass jede Seite das gleiche Design erhält. Dafür kann man CSS-Dateien (externe Stylesheets) einbinden. Externe Stylesheets haben die Endung `.css` und werden mit dem `<link>` Element im `<head>` eingebunden.

HTML `<link rel="..." href="..." media="...">`



rel="..." Ist der Beziehungstyp. Wir verwenden `stylesheet` – das sagt dem Browser, dass es sich um CSS Eigenschaften handelt.

href="..." die Adresse, wo die `.css` Datei zu finden ist.

`href="hauptdesign.css"`

`href="Design/Style.css"`

media="..." Ausgabemedium (z.B. `screen` = Bildschirm, `print` = Druck)



In der externen `.css` Datei, schreibt man gleich wie wir es von der `<head>` Lösung kennen. `<style> ... </style>` ist nicht nötig. Nötig ist aber, eine Regel für unseren Zeichensatz (UTF-8). Eine Regel wird mit dem At-Zeichen (@) eingeleitet.

```
<!doctype html>
<html lang="de">
  <head>
    <link rel="stylesheet" href="stylesheet.css" media="screen">
    ... </head>
```

```
@charset "UTF-8";
/* CSS Document */
body {margin: 0px;}
```

Einfache Selektoren

Man kann auch mehreren Selektoren die gleichen Eigenschaften zuweisen. Diese werden dann durch einen Beistrich getrennt. In folgendem Beispiel wird der Klasse `.tipp` und der Klasse `.hinweis` und dem `<p>` Tag eine blaue Schriftart zugewiesen.

```
.tipp, .hinweis, p {color: blue;}
```

```
<h1 class="tipp">Denkspiele</h1>
<h2 class="hinweis">Online oder Offline</h2>
<p>Es gibt so viele Möglichkeiten</p>
```



Ein HTML Element kann mehrere Klassen haben. Hier im Beispiel hat das `<h1>` Element eine Klassenzugehörigkeit zu `.wichtig` und zu `.fett`.

```
h1 {font-size: larger;}
#haupt {font-family:Arial, Helvetica, sans-serif;}
.wichtig {background:#CCC;}
.fett {font-weight:bolder;}
```

```
<h1 id="haupt" class="wichtig fett">Dokumente</h1>
```



Um alle Elemente anzusprechen, kann man den **Universalselektor** verwenden: Das Sternzeichen `*` Im Beispiel, wird alles in grüner Schriftfarbe geschrieben.

```
* {color: green;}
```

Lernhandout 7.1 Hintergrund

Referenzcode: HCL071

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... nutzen CSS um in ein HTML Element ein Hintergrundbild einzubinden.	I	II
... verstehen die Idee von Ebenen (z. B. Photoshop, zwei Bilder übereinander).	I	I
... skalieren Hintergrundbilder.	I	II
... kacheln (wiederholen) Hintergrundbilder.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- Wenn man zwei Hintergrundbilder übereinander leben möchte, dann sollte man nur Bildformate mit transparentem Hintergrund (wie z. B. PNG) verwenden. Bei zwei JPG Graphiken wird das untere überdeckt.

Wir können schon die Hintergrundfarbe eines Elements bestimmen. Jetzt wollen wir Bilder als Hintergründe verwenden. Dafür gibt es die `background` Eigenschaft.

Hintergrundgrafik einbinden

CSS `background-image: url (" [pfad] ") ;`



[pfad] Es wird der Pfad (absolut oder relativ) zum Bild angegeben. Bei einer externen .css Datei ist der Speicherort der .css Datei ausschlaggebend.

```
#haupt {background-image: url("../pix/hintergr.jpg");}
```



Gibt man zwei `url` Pfade an, dann werden die Bilder übereinander gelegt.

Hintergrundbild skalieren (Größe verändern)

CSS `background-size: [value] ;`



[value] Längenangaben wie z. B. px, % sind möglich. Dabei wird zuerst die Breite und dann die Höhe definiert.
`background-size: 200px 400px;`
 Wird nur ein Wert angegeben, dann wird die Höhe unter Beibehaltung des Seitenverhältnis skaliert.

Folgende Schlüsselwörter sind möglich:

<code>auto</code>	Originalgröße des Bildes, keine Skalierung
<code>contain</code>	Beibehaltung des Seitenverhältnis, die größere Seite wird angepasst.
<code>cover</code>	Beibehaltung des Seitenverhältnis, die kleinere Seite wird angepasst. Damit wird der Anzeigebereich vollständig gefüllt.

```
#haupt {background-size: cover;}
.bilder {background-size: 300px;}
```

Hintergrund wiederholen

CSS `background-repeat: [value] ;`



[value] Wenn das Bild kleiner als das Element ist, kann man es wiederholen (kacheln). Dabei wird der gesamte zur Verfügung stehende Platz ausgefüllt. Folgende Werte sind möglich:

<code>repeat</code>	Wiederholung des Hintergrundbildes
<code>repeat-x</code>	Das Bild wird horizontal wiederholt.
<code>repeat-y</code>	Das Bild wird vertikal wiederholt
<code>no-repeat</code>	keine Wiederholung
<code>space</code>	Das Hintergrundbild wird wiederholt ohne dass ein Bild beschnitten wird.
<code>round</code>	Genauso wie <code>space</code> , hier wird aber das Bild skaliert.

```
#haupt {background-repeat: repeat-x;}
.bilder {background-repeat: space;}
```


Lernhandout 7.2 Hintergrund

Referenzcode: HCL072

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... positionieren Hintergrundbilder in einem HTML Element.	I	II
... verändern den Ausgangspunkt für den Hintergrund.	I	II
... befestigen Hintergrundbilder (verhalten beim Scrolling).	I	II
... definieren Hintergrundeigenschaften für bestimmte Bereiche eines HTML Elements.	I	II
... finden im Internet lizenzfreie Hintergrundbilder.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Position der Hintergrundgrafik

CSS `background-position: [value-x] [value-y] ;`



[value-x] Bestimmt die Position entlang der x-Richtung (horizontal).
Erlaubt sind negative und positive Längenmaße und die Werte:
left center right.

[value-y] Bestimmt die Position entlang der y-Richtung (vertikal).
Erlaubt sind negative und positive Längenmaße und die Werte:
top center bottom.

Der Ausgangswert ist 0 0 – Das ist in der linken oberen Ecke des Elements.

```
#haupt {background-position: 50px 100px;}
```



Mit background-origin wird ein Ausgangspunkt festgelegt.
z. B: background-origin: border-box; richtet den Hintergrund auf die
Außenkante des Rahmens.

Hintergrundbild befestigen

CSS `background-attachment: [value];`



Damit definiert man das Verhalten, wenn ein Element oder sein Inhalt bewegt wird.
Folgende Werte [value] stehen zur Verfügung:

fixed **Bewegung des Elements:** ein anderer Bereich des Hintergrund wird
sichtbar.

Bewegung des Inhalts: der Hintergrund bleibt unverändert.

scroll **Bewegung des Elements:** Hintergrund bleibt unverändert und scrollt
mit dem Element mit.

Bewegung des Inhalts: der Hintergrund bleibt unverändert.

local **Bewegung des Elements:** Hintergrund bleibt unverändert.

Bewegung des Inhalts: der Hintergrund bewegt sich mit.

```
body {background-attachment: fixed;}
```

Hintergrundeigenschaften für bestimmte Bereiche

CSS `background-clip: [value];`



Die Hintergrundeigenschaften (z. B. Bild oder Farbe) werden hier für einen be-
stimmten Bereich festgelegt. Dafür gibt es folgende Werte [value] :

content-box Hintergrund wird nur im Inhalt angezeigt.

padding-box Hintergrund im Inhalt und im Innenabstand

border-box Hintergrund im Inhalt, Innenabstand und im Rahmen.

Der Ausgangswert ist 0 0 – Das ist in der linken oberen Ecke des Elements.

```
#haupt {background-clip: content-box;}
```



Web-Tipp: Auf <http://bg.siteorigin.com/> gibt es einen schönen Background Image
Generator für Hintergrundbilder.

Lernhandout 7.3 Farbverlauf

Referenzcode: HCL073

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... wissen was eine CSS Funktion ist und kennen die Syntax.	I	I
... erstellen einfache Farbverläufe (mit zwei Farben) für HTML Elemente.	I	II
... erstellen komplexe Farbverläufe (mit mehrere Farben).	I	II
... verändern die Erscheinung des <body> Tag.	I	II
... erstellen nur mit einem CSS Farbverlauf eine Österreich Flagge und verwenden dafür Längenangaben.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Um einen Farbverlauf im Hintergrund eines Elements zu erstellen, kann man eine Graphik bzw. Bilddatei erstellen oder man nutzt CSS mit der Funktion `linear-gradient()`. Diese Funktion lässt sich z. B. mit der CSS Eigenschaft `background-image` darstellen.

CSS

```
background-image:
linear-gradient([to value], [farbeVon], [farbeBis]);
```



[to value] Gibt die Richtung an. Dafür kann man ein Winkelmaß (z. B. 75deg) angeben oder in Verbindung mit dem Schlüsselwort `to` die Werte `top`, `right`, `bottom`, `left`. Der Standardwert (ohne Angabe) ist `to bottom = 180deg = 200grad`.

[farbeVon] Der Color-Start-Wert kann eine beliebige Farbangabe sein. Es ist auch der Wert `transparent` möglich. Zusätzlich kann noch ein Längenmaß mitgegeben werden.

[farbeBis] Für den Color-End-Wert gelten die gleichen Bedingungen wie für den Color-Start.

Die Wertangaben werden durch einen Beistrich getrennt.

!!!Achtung!!!: Zwischen `linear-gradient` und `(to left, ...)` darf kein Leerzeichen gesetzt werden.

```
#dahinter {background-image:
            linear-gradient(to top right, red, transparent);}
```



Natürlich kann man mehrere Farben zuweisen.

Im Beispiel: 75° rot bis orange bis gelb bis #FFC bis Weiß.

```
#hinten {background-image:
        linear-gradient(75deg, red, orange, yellow, #FFC, white);
        width:21cm; height:2cm; margin:auto; border: 4px groove;}
```

```
<div id="hinten">&nbsp;  </div>
```



Einen Hintergrundverlauf für den Body-Tag.

```
body {background-image: linear-gradient(to bottom left, blue, white);
      background-attachment:fixed;}
```



Mit CSS eine Österreich-Flagge erstellen.

Dafür benötigen wir Längenangaben.

```
#austria {background-image:
        linear-gradient(red 100px, white 100px, white 200px, red 100px);
        width:400px; height:300px; margin:auto; border: 4px groove;}
.schrift {margin-top:125px; font-size:50px; text-align:center;}
<div id="austria"><p class="schrift">Österreich</p></div>
```

Übungsblatt 7.3 Hintergrund

Referenzcode: HCU073

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... arbeiten strukturiert nach Vorgaben und mit einem fremden HTML Dokument sowie mit bereitgestellten Bildern.	I	II
... binden Hintergrundbilder ein und weisen diesen div. CSS Eigenschaften zu.	I	II
... erstellen CSS Farbverläufe.	I	II
... verspüren ein "London-Feeling".	A	III
Übung B		
... definieren einen CSS Farbverlauf für einen Schriftzug.	I	II
... orientieren sich an dem Screenshot des Übungsblattes.	K	I
... wissen was ein <code>-webkit-</code> ist.	I	I

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- 📄 Zu Übung A Canary Wharf: Damit sollte ein pro europäischer Impuls gesetzt werden und wir gedenken Großbritannien als EU-Mitglied. (stand 2018, mitten im Brexit). England, wir werden dich vermissen.
- 📄 Rainbow Warrior ist der Name des Flaggschiffes von Greenpeace.



Übung A: Canary Wharf

- ☐ Öffne die HTML Datei `canary_wharf.html` (in dieser HTML Seite sind zwei divs. Gekennzeichnet als `ERSTES DIV` und `ZWEITES DIV`).
- ☐ Im Seitenhintergrund soll das Bild `london.png` eingebettet werden. Das Hintergrundbild soll am unteren Seitenrand fixiert und auf die Seitenbreite skaliert dargestellt werden.
- ☐ Ändere die Farbe des Seitenhintergrund in ein angenehmes Blau.
- ☐ `ERSTES DIV`: Suche im Internet nach einem Bild der Englischen Fahne (Union Jack) und binde es als Hintergrundgraphik im ersten Div ein.
- ☐ `ZWEITES DIV`: Definiere einen Farbverlauf, von Oben nach Unten und von Gelb nach Transparent. Entferne den Rahmen um das zweite Div.
- ☐ Überprüfe ob in den Meta-Tags des Dokuments dein Name steht!



Übung B: Rainbow Warrior

- ☐ Definiere einen linearen Farbverlauf (in Regenbogenfarben) für den Schriftzug **RAINBOW WARRIOR**.
- ☐ Du brauchst zusätzlich noch folgende CSS Eigenschaften:
 - `-webkit-background-clip: text;`
 - `-webkit-text-fill-color: transparent;`

ACHTUNG: `-webkit` sind CSS Erweiterungen die für die Aufnahme in die CSS Spezifikationen vorgeschlagen sind. Sie müssen also nicht auf jedem Browser funktionieren.

RAINBOW WARRIOR

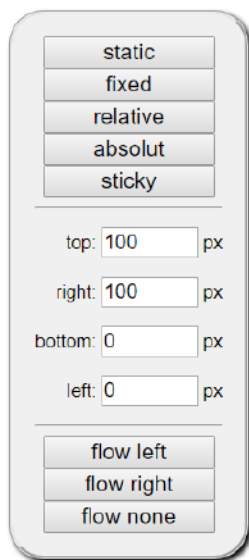
Lernhandout 8.1 Positionierung

Referenzcode: HCL081

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... positionieren Bilder und HTML Elemente im Viewport.	I	II
... kennen und nutzen die unterschiedlichen Positionierungsarten (static, fixed, relative, absolute, sticky).	I	II
... verstehen top, bottom, left und right im Zusammenhang mit der position Eigenschaft.	I	I
... erstellen fixe Fußzeilen (am Browserbottom).	I	II
... erstellen Bildbeschriftungen.	I	II
... kennen die Idee von Ebenen (z. B. wie bei Photoshop) und nutzen den z-index um diese zu bestimmen.	KI	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen



- Im HTML Dokument (Beispiel_Pos.html) kann man die Auswirkungen von position und float ansehen. Links befindet sich eine "Konsole" und rechts im Text ein gelbes Rechteck, welches die Änderungen sofort visualisiert. Für diese Seite wurde Javascript verwendet.

HTML Elemente müssen nicht immer nur in Laufrichtung des Textes positioniert werden. Man kann jedes Element (`div`, Bilder usw.) an jeder Stelle im Browserfenster (viewport) anzeigen. Dafür gibt es die CSS-Eigenschaften `top`, `bottom`, `left` und `right`. Diese können durch numerische Angaben (z. B. `top: 30px;`) bestimmt werden. Zuvor muss man dem Element eine Positionierungsart zuweisen.

CSS `position: [value];`



Für [value] sind folgende Werte möglich:

<code>static</code>	Ist der Defaultwert. Das Element bleibt im Textfluss. <code>top</code> , <code>bottom</code> , <code>left</code> und <code>right</code> werden ignoriert.
<code>fixed</code>	Richtet das Element am Viewport aus und fixiert es dort. Das Element bleibt beim Scrollen an seiner Position. (z. B. eine fixe Kopf- oder Fußzeile).
<code>relative</code>	Richtet das Element von seiner eigenen Position im Textfluss aus. Es hinterlässt eine Lücke.
<code>absolute</code>	Orientiert sich bei <code>top</code> , <code>bottom</code> , <code>left</code> oder <code>right</code> am Elternelement. Jedoch nur, wenn dieses Elternelement selbst mit <code>position: positioniert</code> wurde. Sonst bezieht es sich am <code><html></code> Element. Es behält beim Scrollen also seine Position.
<code>sticky</code>	Behaltet seine Position im Textfluss, bis das obere oder untere Seitenende erreicht wurde. Dort bleibt es dann kleben!

Hier ein Beispiel für eine fixe Fußzeile:

```
#fusszeile {position:fixed;
            bottom:0px; right: 0px;
            font-size: 1.4em; text-align:right;
            width: 100%; background:#F0F0F0;
            padding:10px;}
```

```
<div id="fusszeile">techcom internation GmbH - &copy; 2018</div>
```



Beispiel: Eine Beschriftung über einem Bild!

```
.bild {height:300px; z-index:1;}
.bilddiv {position:relative;}
.beschriftung {position:absolute;
              bottom: 50px; left:10px;
              padding:10px;
              background:white; font-size:1.5em;
              z-index:2;}
```

```
<div class="bilddiv">
  
  <div class="beschriftung">Rail Jet</div>
</div>
```



Wenn Elemente sich überlappen, dann kann man mit `z-index` die Reihenfolge bestimmen, ähnlich wie die Ebenen im Photoshop.

Lernhandout 8.2 display-Eigenschaft

Referenzcode: HCL082

Feinziele Die Schüler innen ...	Zielart	Taxonomie
... verändern die Art eines HTML Elements mit der display Eigenschaft.	I	II
... kennen die unterschiedlichen Werte der display Eigenschaft. (inline, block, inline-block, flex, grid, list-item, table)	I	I
... positionieren mehrere <div> Elemente nebeneinander (mit inline-block, ohne float).	I	II
... unterscheiden zwischen visibility: hidden; und display:none;	I	III
... erkennen den Element- bzw. Textfluss in einem HTML Dokument.	K	III
... zentrieren einen Text horizontal und vertikal über den gesamten Viewport mit display: table-cell;	I	II
... zentrieren einen Text horizontal und vertikal über den gesamten Viewport mit display: flex;	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Mit der `display`-Eigenschaft bestimmt man die Art eines Elements.

CSS `display: [value];`



Für [value] sind folgende Werte möglich:

<code>inline</code>	Inline Boxen verlaufen in einer Zeile entlang der Schreibrichtung. Die Breite und Höhe wird allein durch den Inhalt bestimmt. <code>width</code> und <code>height</code> sind wirkungslos.
<code>block</code>	Block-Boxen haben die gleiche Breite wie das Elternelement. Die Höhe wird durch den Inhalt beeinflusst. Ein Beispiel für eine Block-Box ist der <code><p></code> Tag.
<code>inline-block</code>	Die Breite ist so schmal als möglich. Man kann die Breite mit <code>width</code> festlegen. Die Höhe ist vom Inhalt abhängig, kann aber mit <code>height</code> festgelegt werden.
<code>flex</code>	für flexible Layouts ohne fixe Größen (Flexible Box Layout Module)
<code>grid</code>	für flexible Layouts ohne fixe Größen. Im Gegensatz zu <code>flex</code> , werden hier komplexe Raster erzeugt. (Grid Layout)
<code>list-item</code>	behandelt ein Element wie einen <code></code> Tag.
<code>table</code>	Um ein Element als eine Tabelle darzustellen. In Folge kann ein Block auch als Tabellenzelle mit <code>table-cell</code> dargestellt werden.
<code>none</code>	Das Element wird nicht erzeugt. Es ist unsichtbar und hat keinen Einfluss auf den Elementfluss. (es hinterlässt keine Lücke im Text). <code>display: none;</code> wird oft verwendet wenn man das Medium (z. B. Print) bzw. den Viewport (z. B. für Smartphones) wechselt.

Das Beispiel zeigt, wie man mehrere `<div>` Elemente nebeneinander darstellen kann.

```
.nebenan {display:inline-block;
width:200px; height:200px;
margin:10px; border: 3px groove blue;
text-align: center;}
```

```
<div class="nebenan">EINS</div>
<div class="nebenan">ZWEI</div>
<div class="nebenan">DREI</div>
```



Der Unterschied zwischen `visibility: hidden;` und `display:none;` besteht darin, dass `visibility` das Element ausblendet, während `display: none;` es nicht erzeugt. Der Unterschied liegt dann im Element- bzw. Textfluss.



Zwei Beispiele wie man einen Text horizontal und vertikal über den gesamten Viewport zentriert.

```
#haupt {display:table-cell;
background:#CCC;
height:100vh;
width:100vw;
vertical-align:middle;
text-align:center;}
```

```
#haupt {display:flex;
background:#CCC;
height: 100vh;
align-items:center;
justify-content:center;}
```

Übungsblatt 8.2 Positionierung und Display

Referenzcode: HCU082

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... positionieren HTML Elemente mit CSS.	I	II
... nutzen display-Eigenschaften, wie z. B. inline-block.	I	II
... erstellen eine fixierte Fußzeile.	I	II
Übung B		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... binden mit HTML ein Bild (Screenshot) ein.	I	II
... erstellen eine Beschriftung über das Bild (überlappen).	I	II
Übung C		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen



Übung A: Textgestaltung "Word ..."

- ❑ Öffne die Lösung von **6.2 C: Text mit Spalten** bzw. die Datei Wordelement.html

- ❑ Lösche die Überschrift

```
<h1>Microsoft Word: Einfügen von Elementen</h1>
```

1

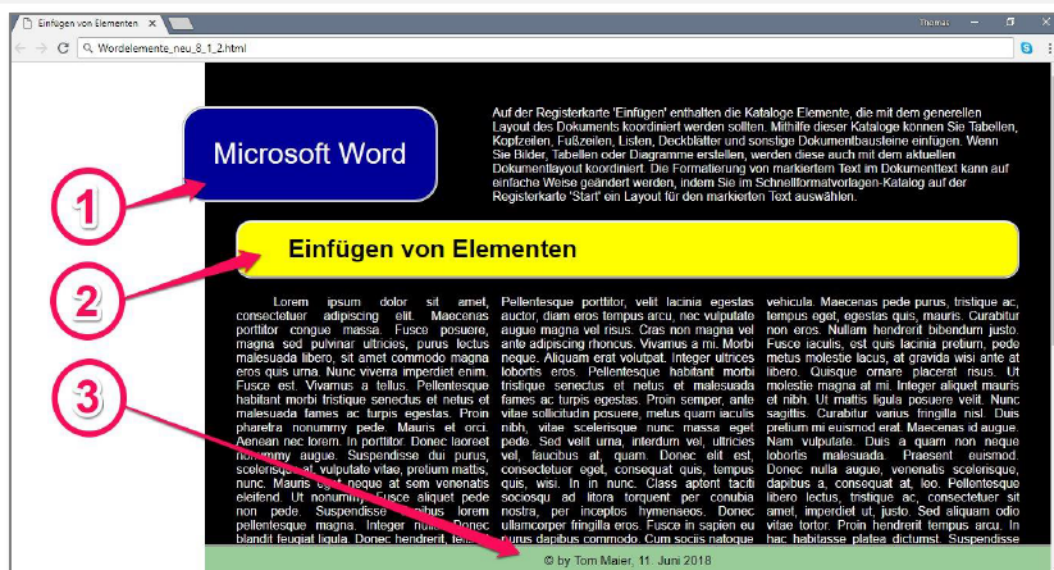
- ❑ Links oben soll ein `<div>Microsoft Word</div>` hinzugefügt werden. Gestalte es ansprechend. Das `<div>` soll nicht über gesamte Bildschirmbreite erscheinen und links über das schwarze `<div id="Hauptdiv">` hinausragen. Beim scrollen soll das `<div>` mitwandern!

2

- ❑ Füge eine weitere Überschrift-Ebene mit dem Text **"Einfügen von Elementen"** hinzu. Die Überschrift soll beim Scrollen am oberen Browserrand 'kleben' bleiben. Gestalte die Überschrift ansprechend!

3

- ❑ Erstelle eine Fußzeile für das HTML-Dokument, die am unteren Browserrand fixiert werden und immer im Vordergrund sein soll. Der Text für die Fußzeile lautet: © by Vorname Nachname, Datum

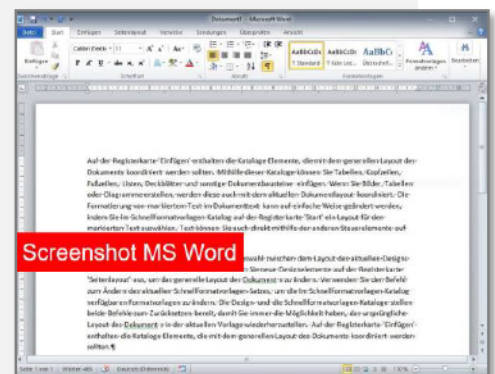


Übung B: Textgestaltung mit Bild

- ❑ Öffne deine Lösung von **8.2 A: Textgestaltung**.

- ❑ Besorge dir einen Screenshot von Microsoft Word und füge ihn als Bilddatei in den HTML Code ein.

- ❑ Beschrifte das Bild mit "Screenshot MS Word". Die Beschriftung soll auf das Bild gelegt werden.



Übung C: Freie kreative Übung

- ❑ Öffne deine Lösung von **8.2 B: Textgestaltung mit Bild** und verändere das Design (andere Schriftarten und Schriftfarben, andere Hintergrundfarben, usw.) damit das Dokument besser und harmonischer aussieht.

Lernhandout 8.3 Strukturierung

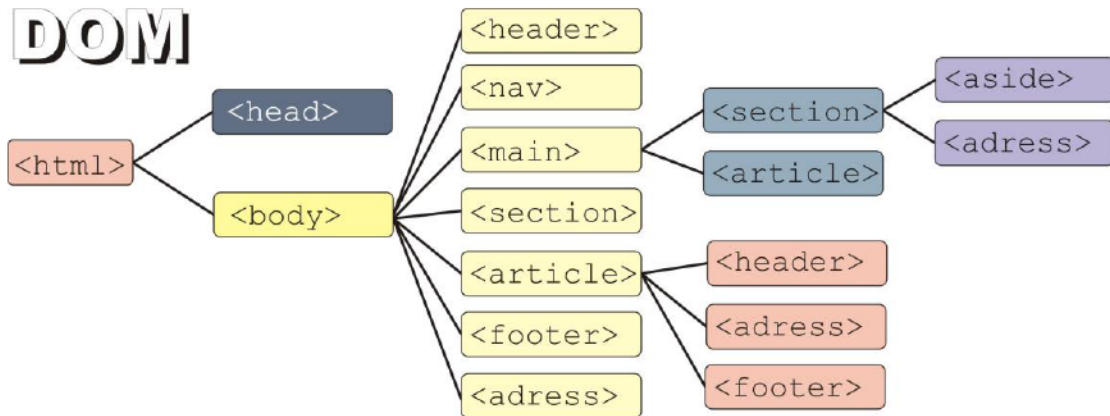
Referenzcode: HCL083

Feinziele Die Schüler innen ...	Zielart	Taxonomie
... verstehen das Document Object Model (DOM).	I	I
... strukturieren HTML Seiten und erstellen damit u. a. barrierefreie Webseiten.	I	II
... definieren die Hauptnavigation mit <nav>.	I	II
... definieren den Hauptinhalt mit <main>.	I	II
... definieren einen Kopfbereich mit <header>.	I	II
... definieren den Schlussbereich für Hinweise (z. B. Impressum) mit <footer>.	I	II
... definieren semantische Gliederungen mit <section> und <article>.	I	II
... definieren Kontaktinformationen mit <address>.	I	II
... definieren Randbemerkungen mit <aside>.	I	II
... erkennen, dass die meisten Tags zur Strukturierung nur geringfügige Auswirkung auf das Aussehen haben aber wichtig für die Semantik der HTML Seite sind.	I	III

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Für die Seitenstruktur haben wir bisher mit `<body>`, `<div>`, `<p>` usw. verwendet. HTML5 bietet aber noch weitere Tags zur Strukturierung. Alle Tags müssen noch mit CSS formatiert werden. Die Strukturierung ermöglicht auch ein barrierefreies Web, weil Screenreader `<header>`, `<nav>` oder `<footer>` überspringen und nur die wichtigen Inhalte vorliest.



HTML `<nav> ... </nav>`



Das `<nav>` Element bezeichnet die Hauptnavigation und bildet einen eigenen Inhaltsbereich.

HTML `<main> ... </main>`



Das `<main>` Element enthält den Hauptinhalt der Webseite.

HTML `<header> ... </header>`



Das `<header>` Element wird für den sichtbaren Kopfbereich verwendet.

HTML `<footer> ... </footer>`



Das `<footer>` Element enthält Informationen zur Website und steht am Ende. Beispiele für Infos im `<footer>` sind Hinweise zu Urheberrecht, Autor oder ein Link zum Impressum.

HTML `<section> ... </section>` und `<article> ... </article>`



Das `<section>` Element umspannt eine thematische Gruppierung. Es dient zur semantischen Gliederung von Inhalten. `<article>` ist mit einem Zeitungsartikel vergleichbar und stellt einen sich geschlossenen Abschnitt dar.



Unter Semantik versteht man die Bedeutung bzw. den Inhalt eines Wortes, Satzes oder Textes.

HTML `<adress> ... </adress>` und `<aside> ... </aside>`



Das `<adress>` Element enthält Kontaktinformationen (z. B. eMail-Adressen, Postadressen, Link zu Seiten mit Kontaktinformationen usw.).

`<aside>` dient für Randbemerkungen. Die Darstellung muss über CSS festgelegt werden.

Lernhandout 8.4 Float

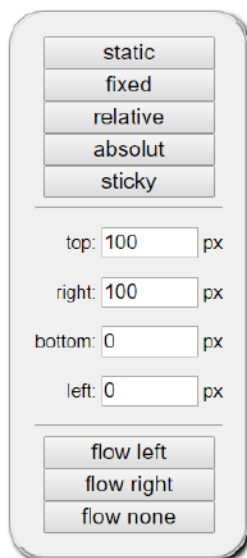
Referenzcode: HCL084

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... verstehen den Text- und Elementfluss von HTML.	I	II
... verschieben Elemente nach links- bzw rechts und lassen den Text umherfließen. (CSS Eigenschaft float).	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
... begreifen den collapse Effekt. (Kindelemente ragen über das Elternelement hinaus).	I	III
... wirken gegen den collapse Effekt mit clear:both;	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen



- Im HTML Dokument (Beispiel_Pos.html) kann man die Auswirkungen von Position und float ansehen. Links befindet sich eine "Konsole" und rechts im Text ein gelbes Rechteck, welches die Änderungen sofort visualisiert. Für diese Seite wurde Javascript verwendet.

Alle HTML Elemente sind im Fluss. Um nun aber Bilder oder andere Elemente links oder rechts vom Fließtext darzustellen, gibt es die *Float* Eigenschaft. Dabei wird das Objekt links oder rechts an die Innenkante des Elternobjekts verschoben.

CSS `float: [Value];`



Für `Value` sind folgende Werte erlaubt:

none	Standardwert, das Element wird nicht verschoben
left	Das Element wird nach links verschoben.
right	Das Element wird nach rechts verschoben.
inherit	Erbt die <code>Float</code> -Eigenschaft des Elternelements.

```
<p>
... weiterer Lauftext ...</p>
```

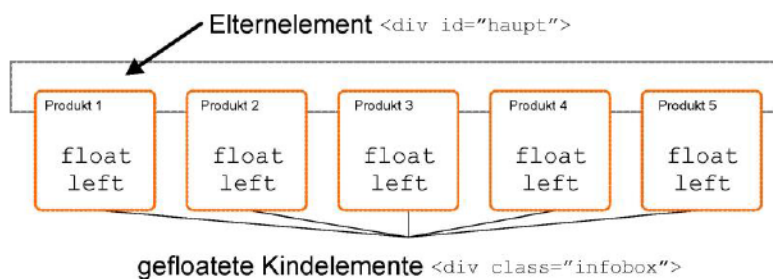


Wenn man mehrere Elemente mit einer `float` Eigenschaft einbindet, dann sind die Kindelemente manchmal größer als das Elternelement und ragen über dieses hinaus. Dieser Effekt wird auch *collapse* genannt. Der Grund: Die Kindelemente werden wegen der `float` Eigenschaft aus dem Fließtext herausgenommen obwohl sie den Text verdrängen!

```
#haupt {border:2px dashed gray; min-height:50px;}
```

```
.infobox {float:left; margin:10px; height:200px; width:200px;
border:2px dotted orange;}
```

```
~~~~~
<div id="haupt">
  <div class="infobox">Produkt 1</div>
  <div class="infobox">Produkt 2</div>
  <div class="infobox">Produkt 3</div>
  <div class="infobox">Produkt 4</div>
  <div class="infobox">Produkt 5</div>
</div>
```



Um dem entgegenzuwirken gibt es mehrere Möglichkeiten. Eine davon ist es, am Ende ein leeres HTML Element mit der CSS Eigenschaft `clear:both;` hinzuzufügen.

```
<div id="haupt">
  ...
  <div class="infobox">Produkt 5</div>
  <div style="clear:both;">&nbsp;</div>
</div>
```

Lernhandout 9.1 Pseudoklassen (Buttons)

Referenzcode: HCL091

Feinziele Die Schüler innen ...	Zielart	Taxonomie
... kennen Syntax und Funktion von dynamischen Pseudoklassen.	I	I
... erstellen Buttons und Schaltflächen mit dynamischen Pseudoklassen.	I	II
... kennzeichnen Hyperlinks mit der Pseudoklasse :link.	I	II
... kennzeichnen besuchte Links mit der Pseudoklasse :visited.	I	II
... kennzeichnen alle Hyperlink-Zustände mit der Pseudoklasse :any-link.	I	II
... kennzeichnen ein Mouse-Over mit der Pseudoklasse :hover.	I	II
... kennzeichnen den Fokus (z. B. durch die Tabulatortaste) mit der Pseudoklasse :focus.	I	II
... kennzeichnen aktive Hyperlinks mit der Pseudoklasse :active.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Um elegante Buttons mit HTML und CSS zu erstellen, kann man dynamische Pseudoklassen verwenden. Sie werden mit einem Doppelpunkt an die CSS Eigenschaft angeschlossen. z. B: `a:link {...}` `.imlink:hover {...}` `#derlink:visited {...}`
Hier in Verbindung mit dem `a href` Element (Hyperlink):

CSS `a:link {...}`

Kennzeichnet nur unbesuchte Links. Die Verwendung ist in den `a-`, `area-` oder `link-` Elementen mit `href`-Attribut möglich.

```
a:link {text-decoration: none;}
```

CSS `a:visited {...}`

Kennzeichnet besuchte Links (also der User hat darauf geklickt). Achtung: Aus Datenschutzgründen unterstützen moderne Browser nur eine eingeschränkte Möglichkeit von Eigenschaften für diese Pseudoklasse!

```
a:visited {text-decoration: line-through;}
```

CSS `a:any-link {...}`

Kennzeichnet besuchte und unbesuchte Links. Es ist also gleichzusetzen mit: `a:link, a:visited {...}`

```
a:any-link {font-weight:bold;}
```

CSS `a:hover {...}`

Kommt zur Anwendung, wenn der Mauszeiger das Element berührt. Verlässt der Mauszeiger das Element, dann wird der vorherige Zustand wieder hergestellt. Die Pseudoklasse `:hover` lässt sich auf beinahe alle HTML Elemente anwenden.

```
a:hover {color:red;}
```

CSS `a:focus {...}`

Sobald der Fokus (z. B: durch die Tabulatortaste) auf ein HTML Element mit einer `:focus` Pseudoklasse gesetzt wird.

```
a:focus {border: 2px solid green;}
```

CSS `a:active {...}`

Sobald der Fokus (z. B: durch die Tabulatortaste) auf ein HTML Element mit einer `:focus` Pseudoklasse gesetzt wird.

```
a:focus {border-bottom: 2px dotted blue;}
```



Hier ein Beispiel für einen Button

```
.schalter:any-link {text-decoration:none; font-size:x-large; padding:10px;
                    border:2px groove gray; border-radius:10px;
                    background:LightGray; color:black;}
.schalter:focus, .schalter:hover, .schalter:active {background:blue;
                                                    color:white;}
```

```
<a href="kontakt.html" class="schalter">Kontakte</a>
<a href="about.html" class="schalter">Über uns</a>
```


Lernhandout 9.2 Pseudoklassen

Referenzcode: HCL092

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... definieren aktive Verweise mit der Pseudoklasse :target.	I	II
... kennen Sprungmarken und Ankerpunkte.	I	I
... können Elemente über einen Link aus- bzw. einblenden.	I	II
... schreiben und verstehen ein umfangreiches Beispiel für --- mehr lesen --- Schaltflächen.	I	II
... kennen weitere strukturelle und dynamische Pseudoklassen.	I	I

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Mit der Pseudoklasse `:target` werden Eigenschaften eines gerade aktiven Verweises dargestellt. Den aktiven Verweis sieht man in der URL Leiste

`http://html.css4.at#L91` öffnet die Webseite und springt zum Element mit der `id="L91"`. Man spricht hier auch von Anker oder Sprungmarken.



Weitere Sprungmarken:

`href="#"` Springt zum Seitenanfang
`href="#top"` Springt ebenfalls zum Seitenanfang
`href=""` erzwingt ein Neuladen der Seite!

CSS :target {...}



Zur Aktivierung ist die Vergabe einer ID notwendig.
 Die Pseudoklasse eignet sich besonders gut in Verbindung mit einer Klasse.



Beispiel: Ein- und ausblenden eines Elements.

```
.details {display: none; background:#F0F0F0; padding:2em;}
.details:target {display:inline-block;}

<h2 id="ueber">Nachrichten</h2>

<p>Die USA plant ein Freihandelsabkommen ...
  <a href="#d1">--- mehr lesen ---</a></p>

<div id="d1" class="details">
  <h3>Weltweiter freier Handel</h3>
  <p>Die USA und die EU haben sich geeinigt. Die Strafzölle werden
    wiederabgeschafft.</p>
  <a href="#ueber" title="schließen">schließen</a>
</div>

<p>Frieden in Korea
  <a href="#d2">--- mehr lesen ---</a></p>

<div id="d2" class="details">
  <h3>Koreas Atompolitik hat positive Folgen</h3>
  <p>Der Konflikt zwischen Nord- und Südkorea ist jetzt
    entgültig zu Ende.</p>
  <a href="#ueber" title="schließen">schließen</a>
</div>
```



Speichere den Code als `nachrichten.html` ab und öffne das Dokument direkt zum Ankerpunkt `d1`. `nachrichten.html#d1`



Es gibt noch eine Vielzahl von strukturellen und dynamischen Pseudoklassen wie z. B.:

`:first-child` Spricht das erste Kindelement an
`:last-child` Spricht das letzte Kindelement an
`:empty` Spricht leere Elemente an (z. B. ohne Text)

Übungsblatt 9.2 Pseudoklassen

Referenzcode: HCU092

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... erstellen dynamische Buttons mit Pseudoklassen.	I	II
... antizipieren zur Lösung über die graphische Darstellung.	I	III
Übung B		
... bearbeiten ein fremdes HTML Dokument.	I	II
... arbeiten nach Vorgaben.	I	II
... verändern die Hyperlinks im Dokument.	I	II
Übung C		
... erstellen eine horizontale Navigationsleiste.	I	II
... definieren Sprungmarken für einzelne Navigationspunkte.	I	II
... fügen dynamische Effekte mit Pseudoklassen hinzu.	I	II
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
Übung D		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... blenden <div> Elemente ein bzw. aus.	I	III

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

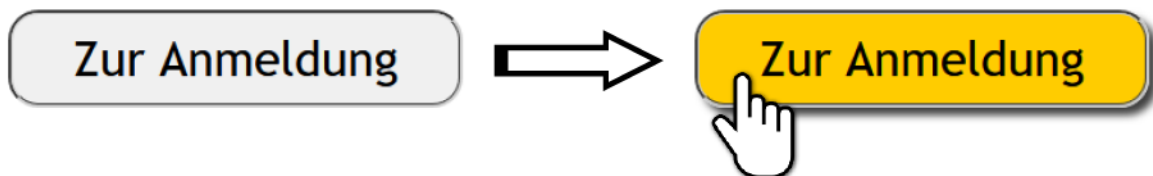
Anmerkungen

- Die Übungen B, C und D sind ein kleiner Beitrag zum Unterrichtsprinzip Umweltschutz.



Übung A: Buttons

- ☐ Gestalte eine Button-Klasse (wie unten dargestellt).
- ☐ Füge die Pseudoklassen (`:link`, `:visited`, `:any-link`, `:hover`, `:focus`, `:active`) hinzu.



Übung B: Links neu gestalten

- ☐ Öffne das HTML Dokument `abfalltrennung.html`
- ☐ Alle Links (`a href`) sollen folgendes Aussehen haben:
- ☐ Jeder Link: keine Unterstreichung, dunkelgraue Schriftfarbe, Fettschrift
- ☐ Besuchte Links: Dunkelgrüne Schriftfarbe
- ☐ Hover: gepunktete Unterstrichen, blaue Schriftfarbe
- ☐ Fokus: einfach Unterstrichen, rote Schriftfarbe
- ☐ Aktive Links: doppelt Unterstrichen, schwarze Schriftfarbe



Übung C: Navigationsleiste

- ☐ Öffne deine Lösung von **9.2 B: Links neu gestalten** (`abfalltrennung.html`).
- ☐ Erstelle eine horizontale Navigationsleiste.
- ☐ Die einzelnen Elemente sollen einen `:hover` und einen `:focus` Effekt haben.
- ☐ Folgende Navigationspunkte sind vorgesehen:
Papier | Leichtverpackung | Metall | Bio | Restmüll
- ☐ Definiere für die Navigationspunkte Sprungmarken (bzw. Anker) zu den dazupassenden Müllsorten.
- ☐ Freie kreative Übung für die Gestaltung und Aussehen!



Übung D: Ein- und ausblenden

- ☐ Öffne deine Lösung von **9.2 C: Navigationsleiste** (`abfalltrennung.html`).
- ☐ Blende alle `<div class="sorte">` mit CSS aus.
- ☐ Per Klick auf einen Navigationspunkt soll das dazupassende `<div class="sorte">` wieder eingeblendet werden.

Lernhandout 9.3 iframe

Referenzcode: HCL093

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... nutzen iFrames um ein "Objektfenster" in den HTML Code einzubauen.	I	II
... kennen und nutzen die Attribute eines iFrame.	I	II
... können YouTube Videos in ein HTML Dokument einbetten.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- 📄 Zur Zeit (HTML5 und CSS3, 2018) sind iFrames eher kritisch zu betrachten. Die Verwendung von <object> Elementen sind definitiv besser, aber da Google seine Maps und Kalender als iFrame anbietet und von YouTube ebenfalls ein iFrame angeboten wird, kann man dieses wohl doch noch bedenkenlos thematisieren.
- 📄 Auf Frames und Framesets wird gänzlich verzichtet, weil diese nicht mehr Zeitgemäß sind und weil mit iFrames der gleiche Effekt erzielt werden kann.

Das *iframe* Element wird zum Einbinden von HTML-Seiten oder anderen Inhalten (z. B. Bilder) verwendet. Dabei werden fremde Quellen eingebunden (innerhalb oder außerhalb der Site).

HTML `<iframe name="" src="" sandbox=""></iframe>`



Das `<iframe>` Element hat folgende Attribute:

height	Höhe (CSS wird empfohlen)
width	Breite (CSS wird empfohlen)
name	Name des iframes (wird benötigt um später über einen Link a href mit dem Attribut target etwas einzubinden).
src	Quelle die beim Start gezeigt werden soll.
sandbox	Sicherheitseinstellungen:
	allow-forms Formularabsendungen sind erlaubt
	allow-popups erlaubt Popups
	allow-scripts erlaubt Scripting
	...

```
<h1>Österreichische online Tageszeitungen</h1>
<ul>
  <li><a href="https://kurier.at/" target="mein_iframe">Kurier</a></li>
  <li><a href="https://www.krone.at/" target="mein_iframe">
    KronenZeitung</a></li>
  <li><a href="https://www.kleinezeitung.at/" target="mein_iframe">
    Kleine Zeitung</a></li>
</ul>

<iframe height="500" width="300" name="mein_iframe"
  src="https://derstandard.at/" sandbox="allow-forms"></iframe>
```



Achtung: Manch fremde Websites verhindert die Darstellung in iFrames mit technischen Hilfsmitteln.



Einbetten eines YouTube Videos!

1: Auf Teilen klicken

Fußball der Philosophie

358.955 Aufrufe



1.438 38 TEILEN

2: Einbetten auswählen

3: Code kopieren und einfügen
Im Jahr 2018 hat YouTube noch iframes verwendet, um das Einbetten von Filmen für eine HTML Seite zu ermöglichen! Es wurden Optionen zum Einbetten angeboten, die aber bei einer W3C Validierung als Fehler angezeigt wurden. Die meisten Browser haben das YouTube Video dennoch korrekt dargestellt.

Embed Video

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/71l
1KytVNCY" frameborder="0"
allow="autoplay; encrypted-media"
allowfullscreen></iframe>
```

☐ Starten bei 0:00

OPTIONEN ZUM EINBETTEN

- ☒ Nach Ende des Videos vorgeschlagene Videos anzeigen.
- ☒ Player-Steuerelemente anzeigen.
- ☒ Videotitel und Player-Aktionen anzeigen.

Übungsblatt 9.3 iframe

Referenzcode: HCU093

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... stellen PDF's in iFrames dar.	I	II
... stellen Überlegungen zur besten Darstellung von DIN A4 Seiten in HTML an.	K	II
... testen die Darstellung in verschiedenen Browsern.	I	II
Übung B		
... bringen eigene Interessen und Vorlieben aus ihrer unmittelbaren Lebenswelt durch ein YouTube-Video zum Ausdruck.	A	III
... binden das YouTube Video mit einem iFrame in ein HTML Dokument ein.	I	II
Übung C		
... beschäftigen sich mit Google Maps.	I	III
... binden eine Google Map in ein HTML Dokument ein.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

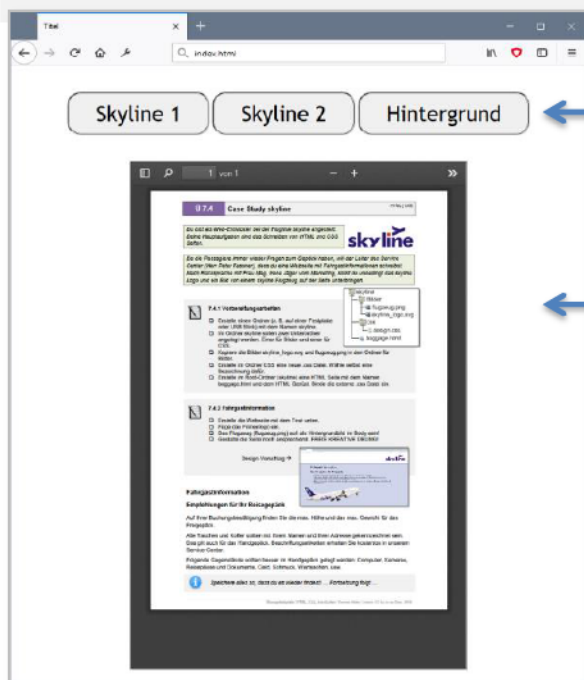
Anmerkungen

- 📄 Zu Übung A: Auf dem Screenshot sieht man ein PDF vom Case-Studie skyline.
- 📄 Zu Übung B und C: Es könnte zu Änderungen auf den externen Websites YouTube und Google kommen.



Übung A: PDF darstellen

- ☐ Erstelle ein neues HTML Dokument
- ☐ Füge ein `iframe` ein mit den ähnlichen Seitenverhältnissen eines DIN-A4 Blattes
TIPP: A4 hat 210 mm Breite und 297 mm Höhe,
(z. B.: Breite: 500 px und Höhe: 707 px)
- ☐ Erstelle drei Link-Buttons.
- ☐ Verlinke die Buttons mit beliebigen PDF Dokumenten (im Din A4 Format). Die Darstellung soll im `iframe` passieren.



Buttons (Target ist das `iframe`)

Die meisten modernen Browser können PDFs darstellen. Einige skalieren die Darstellung brauchbar, einige blenden zusätzlich noch eine Navigation mit ein. Es ist also ratsam die PDF Darstellung in einem `iframe` unbedingt auf verschiedenen Browsern zu testen.



Übung B: YouTube

- ☐ Erstelle ein neues HTML Dokument
- ☐ Suche auf YouTube.com ein cooles Video und bette es in das HTML Dokument ein.



Übung C: Google Map

- ☐ Erstelle ein neues HTML Dokument
- ☐ Suche auf Google Map eine Adresse und bette die Karte in dein HTML Dokument ein.



Hier findest du das Google Maps Menü

Lernhandout 10.2 Pseudoelemente

Referenzcode: HCL102

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... unterscheiden die Syntax von Pseudoklassen und Pseudoelementen.	K	I
... definieren die erste Zeile mit ::first-line	I	II
... definieren den ersten Buchstaben mit ::first-letter	I	II
... fügen vor und/oder nach einem Element einen Content mit ::before und ::after hinzu.	I	II
... definieren die Erscheinung eines markierten Textes mit ::selection.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- Die Schreibweise mit zwei Doppelpunkten ist nicht unbedingt bindend, da die meisten Browser die Pseudoelement auch mit nur einem Doppelpunkt problemlos darstellen können.

Pseudoelemente können an einfache Selektoren (ID, Class usw.) angehängt werden um damit ein Element zu erweitern. Um Pseudoelemente von Pseudoklassen zu unterscheiden, werden sie mit zwei Doppelpunkten (::) geschrieben.

CSS ::first-line {...}



Spricht die erste Text-Zeile eines Elementes an.

```
p::first-line {font-weight:bold;}
```

CSS ::first-letter {...}



Spricht das erste Zeichen in einem Element an.

```
p::first-letter {font-size:4em; float:left; padding:10px;}
```

CSS ::before {content: ...}



Fügt für ein Element zusätzliche Inhalte hinzu.

```
span::before {content: ' OK '};
```

Vor jedem wird ein OK hinzugefügt.

```
p::before {content: url("icon.png");}
```

Vor jedem <p> wird ein PNG-Bild hinzugefügt.

CSS ::after {content: ...}



Analog zu ::before. Der Inhalt wird hier nach dem Element angezeigt.

```
a::before {content: "Link: "};
```

```
a::after {content: " &laquo; "}
```

```
p::after {content: ""; display: inline-block;
background: url("icon.png") no-repeat;
width: 50px; height: 50px;}
```

CSS ::selection {...}



Definiert die Hintergrund- und Schriftfarbe eines markierten Textes. Für Firefox ist die Eigenschaft ::-moz-selection verfügbar.

Das Beispiel zeigt eine Möglichkeit, um das Markieren von Text wenigstens optisch zu unterbinden:

```
div {color:black; background:gray;}
p::selection {color:inherit; background:inherit;}
```

```
<div><p>Sehr viel Lauftext</p></div>
```

Übungsblatt 10.2 Pseudoelemente

Referenzcode: HCU102

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... arbeiten mit einem fremden HTML Dokument.	I	II
... binden Screenshots ein und floaten diese.	I	II
... verändern den ersten Buchstaben (Initial) jedes Absatzes mit CSS.	I	II
... verändern die erste Zeile jedes Absatzes mit CSS.	I	II
Übung B		
... suchen im Internet nach Programmsymbolen.	S	II
... schreiben für jeden Dateityp eine eigene CSS-Klasse.	I	II
... ergänzen die Dateityp-Klassen um die dazu passenden Programmsymbole.	I	II
... ergänzen die Dateityp-Klassen um die dazu passende File-Extension.	I	II
... gestalten mit einer Graphik als Vorlage.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- 📄 Zur Übung A: Die drei Bilder sind eigentlich nicht notwendig. Es ist nur eine Wiederholung des bisher gelernten. Wichtig jedoch sind das Initial und die erste Zeile jedes Absatzes.
- 📄 Zur Übung B: Die Übung soll auf ein Processing vorbereiten (wie z. B. Listen ausgeben mit PHP und MySQL).

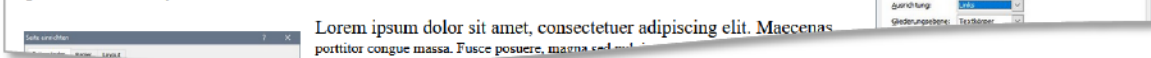


Übung A: Wordelemente

- ☐ Öffne die Webseite **Wordelemente.html**
- ☐ Erstelle drei beliebige Screenshots von Microsoft Word
- ☐ Binde die drei Bilder an beliebiger Stelle ins HTML Dokument ein.
- ☐ Floate zwei Bilder rechts und eines links.
- ☐ Der erste Buchstabe jedes Absatzes soll ein Initial sein. (Hervorgehobener Buchstabe).
- ☐ Die erste Zeile jedes Absatzes soll etwas hervorgehoben werden!








Microsoft Word: Einfügen von Elementen

Auf der Registerkarte 'Einfügen' enthalten die Kataloge Elemente, die mit dem generellen Layout des Dokuments koordiniert werden sollten. Mithilfe dieser Kataloge können Sie Tabellen, Kopfzeilen, Fußzeilen, Listen, Deckblätter und sonstige Dokumentbausteine einfügen. Wenn Sie Bilder, Tabellen oder Diagramme erstellen, werden diese auch mit dem aktuellen Dokumentlayout koordiniert. Die Formatierung von markiertem Text im Dokumenttext kann auf einfache Weise geändert werden, indem Sie im Schnellformatvorlagen-Katalog auf der Registerkarte 'Start' ein Layout für den markierten Text auswählen.



Übung B: Datei Explorer

- ☐ Suche im Internet nach Programmsymbolen für
 - Word Dateien (.docx)
 - Excel Dateien (.xlsx)
 - PDF Files (.pdf)
- ☐ Erstelle ein HTML Dokument.
- ☐ Schreibe für jeden Dateityp (Word, Excel, PDF) eine eignen CSS-Klasse.
- ☐ Vor jeder Klasse soll das dazu passende Icon (Programmsymbol) automatisch hinzugefügt werden. Verwende dafür: `::before`
- ☐ Nach jeder Klasse soll die dazu passende Dateierweiterung automatisch hinzugefügt werden. Verwende dafür: `::after`
- ☐ Gestalte einen "Datei Explorer" wie unten dargestellt.

Name	Datum	Typ	Größe
 Planung.docx	2018-06-30	Word-Dokument	30 KiB
 Konzeption.docx	2018-06-30	Word-Dokument	45 KiB
 Machbarkeit.docx	2018-06-29	Word-Dokument	104 KiB
 Kostenrechnung.xlsx	2018-06-29	Excel-Dokument	78 KiB
 Ist_soll_Vergleich.xlsx	2018-06-30	Excel-Dokument	47 KiB
 Endbericht.pdf	2018-06-28	PDF-Dokument	308 KiB
 Best-Case.pdf	2018-06-30	PDF-Dokument	144 KiB

Lernhandout 10.3 Audio

Referenzcode: HCL103

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... stellen Überlegungen zur Verwendung von Multimediaelementen für das Web an.	K	III
... verwenden das <audio> Element um Musik und Soundfiles in ein HTML Dokument einzubinden.	I	II
... kennen und nutzen die Attribute des <audio> Elements.	I	II
... verstehen was ein Fallback ist.	I	I
... definieren ein Fallback für das <audio> Element mit einem <source> Tag.	I	II






Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- Es ist ratsam die Schüler|innen mit Kopfhörern auszustatten, denn wenn das Plenum (jeweils jeder einzelne) die Europahymne abspielt, dann kann es schon zu extrem viel "Freude schöner Götterfunken" kommen. Die Übungsbeispiele sollen dem Unterrichtsprinzip Europa entsprechen.

Mit HTML 5 kann man auch Audio- und Videodateien einbinden. Ratsam ist es, mit Multimediaelementen gut überlegt umzugehen. Automatisch abgespielte Soundfiles können den Benutzer entnerven – große Dateien brauchen selbstverständlich mehr Ladezeit.

Browsersupport					
<audio>	4.0	9.0	3.5	4.0	10.5

HTML <audio> </audio>



Es können Musik und Soundfiles in den Formaten .mp3, .wav und .ogg abgespielt werden. Wobei .mp3 den Vorzug haben sollte, weil dieses Format von den meisten Browsern unterstützt wird.

Folgende Attribute sind für das <audio> Element verfügbar:

autoplay	beginnt sofort mit dem Abspielen ohne Wert
controls	zeigt die Steuerungsfunktion an (Lautstärke usw.) ohne Wert
loop	das Abspielen wird endlos wiederholt ohne Wert
muted	zu Beginn stummschalten ohne Wert
preload	definiert das Ladeverhalten: auto ← die gesamte Datei wird geladen none ← die Datei wird nicht vorgeladen metadata ← es werden nur die Metadaten geladen
src	URL zur Audiodatei

Zusätzlich gibt es noch TimeRange Attribute die für JavaScript von Bedeutung sind:

buffered	liest aus, wieviel zwischengespeichert wurde
played	liest aus, wie viel schon gespielt wurde.

Der Text zwischen dem <audio> Tag wird angezeigt, wenn das Abspielen nicht möglich ist. z. B. weil es in den Browsereinstellungen deaktiviert wurde.

```
<audio controls loop src="Europahymne.mp3">
  Keine Audiowiedergabe möglich!</audio>
```



Ein Fallback ist eine Alternative die Abgespielt werden soll, wenn das primäre Sound-Format nicht abgespielt werden kann.
Dafür wird im <audio> Element weitere <source> Tags setzen:

```
<audio controls autoplay="true" >
  <source src="Europahymne.mp3" type="audio/mp3" />
  <source src="Europahymne.ogg" type="audio/ogg" />
  Keine Audiowiedergabe möglich!
</audio>
```

Übungsblatt 10.3 Audio

Referenzcode: HCU103

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... erstellen eine Webseite und binden Soundfiles ein.	I	II
... beschriften die <audio> Tags mit einem Hinweis.	I	II
... finden für die Soundfiles die passenden <audio> Attribute.	I	III
... definieren einen Fallback.	I	II
... testen das HTML Dokument auf verschiedenen Browsern.	I	II
... arbeiten strukturiert nach Vorgabe.	S	II
Übung B		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

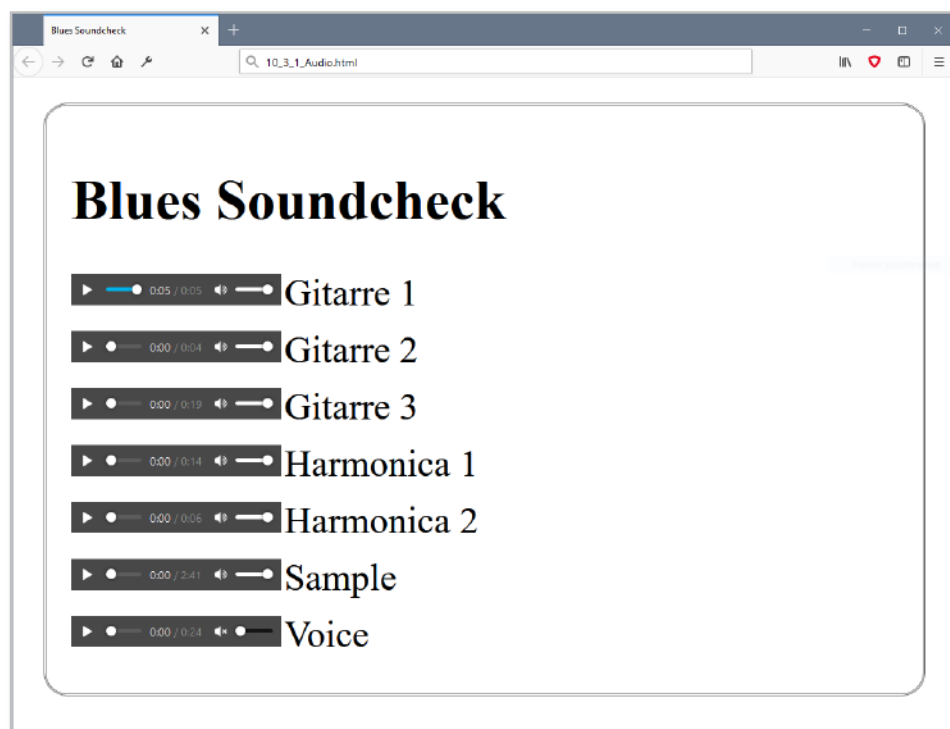
Anmerkungen

- Die Soundfiles aus dem Container Bluesfiles.zip sind akribisch ausgesucht und wurden mit der Common Creative Lizenz CC0 gekennzeichnet.



Übung A: Blues Soundcheck

- ☐ Entzippe die Datei **Bluesfiles.zip**
- ☐ Erstelle ein neues HTML Dokument mit allen Vorarbeiten (Title, Datum usw.)
- ☐ Füge die Überschrift "Blues Soundcheck" hinzu.
- ☐ Füge alle Soundfiles aus der Datei **Bluesfiles.zip** mit dem `<audio>` Tag hinzu.
- ☐ Beschrifte die `<audio>` Tag mit einem Hinweistext falls die Darstellung nicht möglich ist und einer lesbaren Bezeichnung des Audioelements.
- ☐ Alle `<audio>` Elemente haben Bedienelemente (Play, Lautstärke usw.)
- ☐ **guitar.wav** soll beim Öffnen der Webseite sofort abgespielt werden.
- ☐ Erstelle für **guitar3.ogg** ein Fallback auf **guitar3.wav**.
Die Idee: **guitar3.ogg** hat 311 KiB und soll deshalb auch bevorzugt abgespielt werden. Da aber .ogg Files nicht von allen Browsern abgespielt werden, soll alternativ die **guitar3.wav** mit 3369 KiB abgespielt werden.
- ☐ **harmonica.mp3** soll endlos wiederholt werden.
- ☐ **sample.mp3** hat eine Größe von 2.5 MiB und soll deshalb automatisch geladen, aber noch nicht automatisch abgespielt werden!
- ☐ **voice.mp3** soll zum Start stumm geschaltet sein.
- ☐ Teste dein Webdokument auf verschiedenen Browsern!



Übung B: Blues Design

- ☐ Öffne deine Lösung von **10.3 A: Blues Soundcheck** und verpasse dem Dokument ein cooles Blues Design.
- ☐ Freie kreative Übung!

Lernhandout 10.4 Video






Referenzcode: HCL104

Feinziele Die Schüler innen ...	Zielart	Taxonomie
... binden Videodateien in ein HTML Dokument ein.	I	II
... kennen und nutzen die Attribute des <video> Elements.	I	II
... kennen die Videoformate mp4, WebM und Ogg.	I	I
... erstellen ein Fallback und ein Vorschaubild.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Mit HTML 5 kann man Videodateien einbinden ohne das Plugins oder Add-Ons notwendig sind.

Browsersupport					
<audio>	4.0	9.0	3.5	4.0	10.5

HTML <video> </video>



Es können Videos in den Formaten .mp4, .ogg und .WebM abgespielt werden. Wo bei man .mp4 bevorzugen sollte, weil dieses Format von den meisten Browsern unterstützt wird.

Folgende Attribute sind für das <video> Element verfügbar:

autoplay	beginnt sofort mit dem Abspielen ohne Wert
controls	zeigt die Steuerungsfunktion an (Play usw.) ohne Wert
loop	das Abspielen wird endlos wiederholt ohne Wert
muted	den Ton auf stummschalten ohne Wert
preload	definiert das Ladeverhalten: auto ← die gesamte Datei wird geladen none ← die Datei wird nicht vorgeladen metadata ← es werden nur die Metadaten geladen
src	URL zur Videodatei
height, width	Größenangaben (ohne Angaben wird die Größe automatisch erkannt)

```
#myvideo {position:fixed; top:0px; left:0px;
min-height:100%; min-width:100%;}
```

```
<video id="myvideo" src="paint.mp4" autoplay controls >
  Das Video kann nicht dargestellt werden!</video>
```



MP4 = MPEG4-Dateien mit H264 video codec + AAC audio codec
WebM = WebM-Dateien mit VP8 oder VP9 video codec + Vorbis audio codec
Ogg = Ogg-Dateien Theora video codec + Vorbis audio codec



Ein Fallback ist eine Alternative die abgespielt werden soll, wenn das primäre Video-Format nicht angezeigt werden kann. Dafür werden im <video> Element weitere <source> Tags gesetzt. Zusätzlich kann mit dem poster Attribut ein Vorschaubild definiert werden.

```
<video controls autoplay="true" poster="vorschau.png" >
  <source src="paint.mp4" type="video/mp4">
  <source src="paint.ogg" type="video/ogg">
  Keine Videoanzeige möglich!
</video>
```


Übungsblatt 10.4 Video

Referenzcode: HCU104

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... binden ein mp4 Videofile in ein HTML Dokument ein.	I	II
... verwenden die Attribute autoplay und loop.	I	II
... erstellen eine Beschriftung des Videos in Form eines überlappten <div>.	I	II
... erweitern die Beschriftung um einen Hover-Effekt.	I	II
... kennen und nutzen die CSS Eigenschaft opacity.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- Das Beispielvideo paint.mp4 wurde von pixbay.com mit der Common Creative Lizenz CC0 entnommen.



Übung A: Video mit Beschriftung

- ☐ Erstelle eine neue HTML Seite.
- ☐ Binde die Videodatei paint.mp4 in das HTML Dokument ein.
- ☐ Das Video soll automatisch starten.
- ☐ Das Video soll sich ständig wiederholen.
- ☐ Definiere **über dem Video** ein Beschriftungs-Div mit dem Text:
Actionpainting
Auf pixabay.com von Piro4D, Auflösung 640 x 360 Pixel
- ☐ Das Beschriftungs-Div soll unsichtbar sein und erst wenn man mit dem Mauszeiger darüberfährt (Hover-Effekt) erscheinen!








Es gibt mehrere Möglichkeiten, diese Aufgabe zu lösen. Am elegantesten ist wohl die CSS Eigenschaft `opacity`. Diese bestimmt die Transparenz (Durchsichtigkeit) eines HTML Elements.



CSS `opacity: Value;`



Für den `Value` ist ein Zahlenwert zwischen 0 und 1 möglich. Wobei 0 keine Deckkraft und 1 eine 100%ige Deckkraft bedeutet. 0.5 ist halbdurchlässig.

Browsersupport					
<code>opacity</code>	4.0	9.0	2.0	3.1	9.0

Lernhandout 11.1 Medienabfrage

Referenzcode: HCL111






Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... wissen was, Medienabfragen, Medientypen und Medienmerkmale sind.	I	I
... verstehen die Syntax von Medienabfragen.	I	II
... kombinieren Medientypen mit Medienmerkmalen.	I	II
... optimieren Webseiten für den Druck.	I	II
... optimieren Webseiten für unterschiedliche Anzeigeflächen.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Medienabfragen bestimmen die CSS Darstellung für ausgewählte Medientypen (z. B. Bildschirm, Druck) und/oder Medienmerkmalen (z. B. Höhe und Breite des Anzeigebereichs). Die `@media` Abfrage kann überall im CSS Bereich (z. B. im Head oder in einer externen CSS) eingebunden werden.

Browsersupport					
@media	21	9.0	3.5	4.0	9

CSS `@media Medientyp and (Medienfeature) {...};`



Folgende *Medientypen* sind möglich:

<code>all</code>	alle Ausgabemedien (Standardwert)
<code>print</code>	nur für den Druck
<code>screen</code>	nur für Bildschirme
<code>speech</code>	für die Sprachausgabe

Es gibt eine Vielzahl von Medienmerkmalen. Diese *Medienfeatures* gehören zu den wichtigsten:

<code>width</code>	Breite eines Anzeigebereichs, <code>min-</code> und <code>max-</code> sind erlaubt.
<code>height</code>	Höhe eines Anzeigebereichs, <code>min-</code> und <code>max-</code> sind erlaubt.
<code>orientation</code>	Seitenformat eines Mediums. Möglich sind <code>landscape</code> (Querformat) und <code>portrait</code> (Hochformat)
<code>color</code>	Anzahl der Farbbits die ein Ausgabegerät anzeigen kann. Der Wert 0 entspricht einer Anzeige ohne Farben.
<code>resolution</code>	Die Auflösung (Dichte der Bildpunkte) auf dem Ausgabemedium.
<code>scripting</code>	Wenn Scriptsprachen (z. B. JavaScript) verfügbar ist.

```
@media screen and (min-width: 1200px) {
    .schalter {font-size:larger;}
}
```



Wenn man die Webseite für den Ausdruck vorbereiten möchte:

```
@media print {
    #inhalt {font-size: 12pt; width: 18cm; margin:auto;
        background-color: white;}
}
```



Wenn eine Anzeigefläche (z. B. Browserbreite) kleiner-gleich 600px ist:

```
@media screen and (max-width: 600px) {
    .schalter {display:none;}
}
```

Übungsblatt 11.1 Medienabfragen

Referenzcode: HCU111

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... definieren eine Medienabfrage für einen Viewport (Browserbreite) von mehr als 1020 Pixel.	I	II
... definieren eine Medienabfrage für einen Viewport (Browserbreite) mit weniger als 800 Pixel.	I	II
... definieren eine Medienabfrage für einen Viewport (Browserbreite) mit weniger als 600 Pixel.	I	II
... testen die Website mit den Web-Entwickler-Tools von Firefox.	I	II
Übung B		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... nutzen Medienabfragen um Text ein- bzw. auszublenden.	I	IV
Übung C		
... erstellen einen Lebenslauf mit HTML und CSS.	I	II
... optimieren den Lebenslauf für den Druck.	I	IV
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen



Übung A: Media Queries

- ☐ Erstelle ein neues HTML Dokument.
- ☐ Füge ein `<div>` Element mit folgenden CSS Eigenschaften ein:
Breite: 80%, Höhe: 300 Pixel
Zentriert über die Bildschirmbreite
Rahmen: 10 Pixel Groove in der Farbe Blau
Hintergrundfarbe: Grün
- ☐ Wenn die Browserbreite größer als 1020 Pixel ist, soll das `<div>` einen schwarzen Hintergrund haben.
- ☐ Wenn die Browserbreite kleiner als 800 Pixel ist, soll das `<div>` einen blauen Hintergrund haben mit einem orangen Rahmen.
- ☐ Wenn die Browserbreite kleiner als 600 Pixel ist, soll das `<div>` keinen Rahmen mehr haben und eine Breite von 100% einnehmen.
Hintergrundfarbe: Grün.
- ☐ Teste deine Webseite
(z. B. mit Firefox → Web-Entwickler-Tools → Bildschirmgrößen testen).



Übung B: Media Queries mit Text

- ☐ Öffne deine Lösung von oben (11.1 B: Media Queries).
- ☐ Bei einer Browserbreite von mehr als 1020 Pixel soll der Text: "**Größer als 1020 Pixel**" erscheinen.
- ☐ Bei einer Browserbreite von weniger als 800 Pixel soll der Text: "**Kleiner als 800 Pixel**" erscheinen.



Übung C: Lebenslauf

- ☐ Erstelle einen Lebenslauf (von dir oder einer anderen prominenten Persönlichkeit) mit folgenden Inhalten:
 - Persönlichen Daten (Name, Geburtsdatum, usw.)
 - Berufspraxis (Ferialpraktika udgl.)
 - Ausbildungen
 - Besondere Fähigkeiten (HTML, CSS, Buchhaltung usw.)
 - Hobbies (Webseiten schreiben usw.)
- ☐ Optimierte den Lebenslauf für Bildschirmdarstellungen und für den Druck!
- ☐ Freie kreative Übung!

Lernhandout 11.2 Objekte

Referenzcode: HCL112

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... fügen Bilder mit dem <picture> Element ein.	I	II
... kombinieren das <picture> Element mit Medienabfragen.	I	II
... definieren einen Fallback für das <picture> Element.	I	II
... fügen diverse Objekte (z. B. Java Applets, ActiveX, Flash) mit dem <object> Element in ein HTML Dokument ein.	I	II
... kennen die Attribute des <object> Elements.	I	II
... binden Anwendungen oder interaktive Inhalte mit <embed> ein.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Bisher haben wir Bilder mit dem `img` Element eingefügt. Mit dem `picture` Element in Verbindung mit `source` können wir verschiedene Bildversionen für verschiedene Pixeldichten, Auflösungen und Bildschirmformate einbinden. Im `picture` Element werden die Bilder mit dem `source` Tag und dem `img` Element als Fallback angegeben.

HTML `<picture> ... </picture>`



Im Beispiel wird das Bild `Zug_800.jpg` angezeigt, wenn der Anzeigebereich größer als 800 Pixel ist. Bei einem Viewport zwischen 800 und 600 Pixel wird das Bild `Zug_600.jpg` verwendet – beide wurden mit dem `source` Tag eingebunden. Als Fallback dient das `img` Element mit dem Bild `Zug_400.jpg`. Es wird angezeigt, wenn keines der anderen Bilder angezeigt werden kann, bzw. wenn das `picture` Element vom Browser nicht unterstützt wird.

```
<picture>
  <source media="(min-width: 800px)" srcset="Zug_800.jpg">
  <source media="(min-width: 600px)" srcset="Zug_600.jpg">
  <!--Fallback-->
  
</picture>
```

Das `object` Element definiert eingebunden Objekte innerhalb des HTML Dokuments. `object` erlaubt das Einbinden von Audio, Video, Java Applets, ActiveX, PDF und Flash Files. Darüber hinaus, kann man auch ein anderes HTML Dokument oder Bilder anzeigen lassen. Mit dem `param` Tag übergibt man Parameter an das Plug-In.

HTML `<object>...</object>`



Folgende Attribute für `object` sind verfügbar:

<code>data</code>	bestimmt den Ort der Quelle.
<code>form</code>	Identifizierungsname für form Elemente
<code>height</code>	Höhe in Pixel oder %
<code>width</code>	Breite in Pixel oder %
<code>name</code>	Identifizierungsname des Elements
<code>type</code>	der Typ der Quelle

```
<object data="data/test.htm" type="text/html" width="300" height="200">
  Alternativ: <a href="data/test.htm">test.htm</a>
</object>

<object width="400" height="400" data="helloworld.swf"></object>
```

HTML `<embed>...</embed>`



Zum Einbinden von Anwendungen oder interaktiven Inhalten.

```
<embed src="helloworld.swf">
```

Übungsblatt 11.2 Objekte

Referenzcode: HCU112

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... erstellen eine einfache Webseite.	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
Übung B		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... binden eine Flashanimation in das HTML Dokument ein.	I	II
... setzen Ankerpunkte vom Home Button auf die Flashanimation.	I	II
Übung C		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... fügen Bilder mit dem <picture> Elements ein.	I	II
... definieren Medienabfragen für die <source> Elemente innerhalb des <picture> Tags.	I	II
... fügen ein Fallback hinzu.	I	II
... setzen Sprungmarken vom Navigationsbutton zum <picture> Element.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen



Übung A: Smartphone Shop I

- ☐ Entzippe die Datei `Smartphone_Shop.zip`
- ☐ Erstelle ein neues HTML Dokument.
Der `<body>` hat eine schwarze Hintergrundfarbe.
- ☐ Erstelle einen Banner mit der Bilddatei `background.jpg` als Hintergrund.
- ☐ Im Zentrum des Banners steht: "Smartphone Shop Graz"
- ☐ Unter dem Banner soll eine horizontale Navigation sein.
- ☐ In der Navigation sind folgende Buttons:
Home | Produkte | Angebote | Webshop | Kontakt
- ☐ Gestalte die Webseite anspruchsvoll.
- ☐ Speichere die Webseite als `index.html` ab.



Übung B: Smartphone Shop II

- ☐ Öffne dein HTML Dokument `index.html` aus der Übung 11.2 A: Smartphone Shop.
- ☐ Bette die Flashdatei `flashile.swf` ins Dokument ein.
- ☐ Setze eine Sprungmarke/Anker vom **Home Button** auf die Flashdatei.



Übung C: Smartphone Shop III

- ☐ Öffne dein HTML Dokument `index.html` aus der Übung 11.2 B: Smartphone Shop.
- ☐ Füge ein `<picture>` Element ein.
- ☐ Erstes Bild: `phone1.jpg`
- ☐ Zweites Bild: `phone2.jpg` (kleiner-gleich 800 Pixel Browserbreite).
- ☐ Drittes Bild: `phone3.jpg` (kleiner gleich 600 Pixel Browserbreite).
- ☐ `phone3.jpg` soll auch das Fallback-Bild sein!
- ☐ Setze eine Sprungmarke/Anker vom **Produkte Button** auf das `<picture>` Element.

Lernhandout 11.3 Viewport

Referenzcode: HCL113

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... kennen die Probleme und Anforderungen von HTML Seite für mobile Geräte (Smartphone, Tablet).	I	III
... definieren einen Viewport.	I	II
... wissen, was Responsive Web Design bedeutet.	I	I
... nutzen die Vorgaben des RWD (Responsive Web Design) für die Gestaltung von Webseiten um die Darstellung auf allen Geräten zu gewähren.	I	II
... definieren einen Viewport für mobile Geräte.	I	II
... erweitern den Viewport-Content um ein Zoomlevel und Skalierbarkeit.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Natürlich ist es unumgänglich, die Webseiten auch für Mobile Devices tauglich zu machen. Die Webseiten sollen auf Smartphones oder Tablets ebenso schön sein, wie bei einer Browserdarstellung. Dabei gibt es zwei Probleme. Erstens, haben Mobile Devices eine beträchtlich größere Pixeldichte (Auflösung) als ein herkömmlicher Bildschirm und zweitens sind ihre Screens viel kleiner. Das führt dazu, dass der Mobile Browser die Webseite zwar vollständig darstellen kann, aber Schriften, Bilder usw. sind dann sehr klein. Um dem entgegen zu wirken, kann man den Viewport definieren!

Dieser wird im `<head>` als `<meta>` Tag angegeben. Das Beispiel sagt den mobilen Geräten, dass die Seite mit einer Breite von 1024 Pixel dargestellt werden soll:

```
<!DOCTYPE html>
<head>
  <meta name="viewport" content="width=1024" />
</head>
<body>
</body>
```

Responsive Web Design

Responsive Web Design (RWD) ist ein Design und Layout-Konzept um Webseiten auf allen Geräten gut darzustellen. Es werden HTML Elemente in seiner Größe verändert, versteckt, ein- oder ausgeblendet und an eine andere Position gestellt. Dabei sollte nur HTML und CSS verwendet werden!

Für Mobile Devices setzen wir den Viewport auf die Screen-Breite des Gerätes:
(Der Meta-Tag sollte auf jeder Website eingetragen werden!)

```
<meta name="viewport" content="width=device-width">
```



Im Attribut `content` des `<meta>` Tags, können auch das Zoom Level (`initial-scale`) bestimmt werden und ob der/die Nutzer/in die Seite zoomen kann (`user-scalable`).

```
<meta name="viewport"
  content="width=device-width, initial-scale=1.0, user-scalable=no">
```

Der/die User/in soll auf den mobilen Geräten zwar weiterhin vertikal scrollen, jedoch nicht horizontal. Ebenso soll ein Zoomen nicht notwendig sein, um die Inhalte gut zu lesen bzw. betrachten. Dafür ist folgendes zu beachten:

- 1. Verwende keine Elemente mit fixen Größenangaben.**
Wo immer es möglich ist sollte statt `px` besser `%` verwendet werden. Schriftgrößen sind in `em` oder ausgeschrieben (`medium`) empfehlenswerter als in `pt` oder `px`. usw.
- 2. Inhalte sollen nie auf eine bestimmte Darstellungsbreite angewiesen sein, um gut dargestellt zu werden.**
Wir müssen davon ausgehen, dass die Webseite sowohl auf kleinen Screens als auch auf extrem großen Screens aufgerufen werden kann. Auch ist nicht sicher, ob ein Browser immer im Vollbildmodus geöffnet ist. Deshalb sollte man die Elemente am besten im Fluss lassen – wo immer es möglich ist.
Verwende Medienabfragen (media queries) um das HTML Dokument für kleine und große Screens zu optimieren.

Übungsblatt 11.3 Viewport

Referenzcode: HCU113

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... betten ein PDF File in ein HTML Dokument.	I	II
... setzen Ankerpunkte von der Navigation auf das PDF.	I	II
... fügen eine Google-Map hinzu.	I	II
Übung B		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... optimieren eine Webseite für mobile Geräte.	I	II
... wandeln eine horizontale Navigation in eine vertikale Navigation.	I	II
Übung C		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... fügen einen Button zum ein- und ausblenden des Navigationsmenü hinzu.	I	II
Übung D		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... suchen Probleme und stellen Überlegungen zu dieser Webseite an.	KI	III
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen



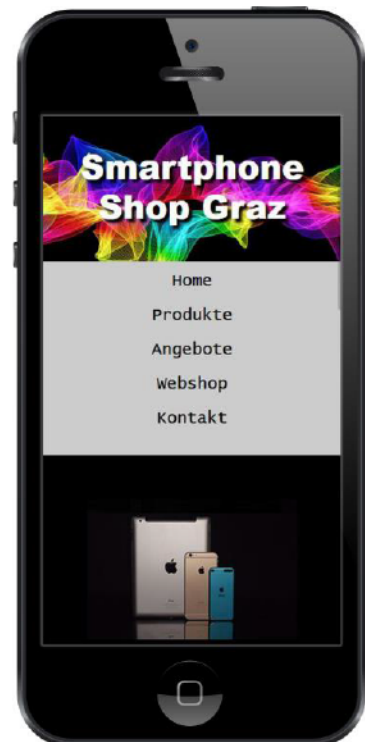
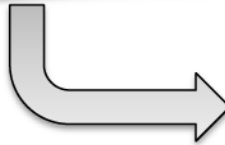
Übung A: Smartphone Shop IV

- ☐ Öffne dein HTML Dokument `index.html` aus der Übung 11.2 A: **Smartphone Shop III**.
- ☐ Bette das PDF-File `Aktion.pdf` ins Dokument ein.
- ☐ Setze eine Sprungmarke/Anker vom **Angebote Button** auf das PDF.
- ☐ Füge folgende Adress-Informationen hinzu:
Smartphone Shop Graz
Jakoministraße 15, 8010 Graz
Telefon: 0316 55587
eMail: shop@smartgraz.at
- ☐ Füge noch ein Google Maps `iframe` mit der Adresse oben hinzu und setze eine Sprungmarke vom **Kontakt Button** auf die Adress-Informationen.



Übung B: Smartphone Shop V

- ☐ Öffne dein HTML Dokument `index.html` aus der Übung 11.3 A: **Smartphone Shop IV**.
- ☐ Die Webseite soll für mobile Geräte optimiert werden (Viewport, Medienabfragen).
- ☐ Die horizontale Navigation soll auf mobilen Geräten vertikal dargestellt werden!



Übung C: Smartphone Shop VI

- ☐ Öffne dein HTML Dokument `index.html` aus der Übung 11.3 B: **Smartphone Shop V**.
- ☐ Blende die Navigation für mobile Geräte aus.
- ☐ Füge einen Button hinzu, mit dem man das Navigationsmenu einblenden kann.



Übung D: Smartphone Shop VII

- ☐ Öffne dein HTML Dokument `index.html` aus der Übung 11.3 C: **Smartphone Shop VI**.
- ☐ Analysiere die Probleme, die sich für diese Webseite ergeben.
- ☐ Verbessere und viel wichtiger, verschönere die Webseite!
- ☐ Freie kreative Übung!

Lernhandout 11.4 Box Eigenschaften

Referenzcode: HCL114

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... bestimmen für eine Box einen Schattenverlauf.	I	II
... kennen die unterschiedlichen Attribute von box-shadow.	I	II
... bestimmen das Verhalten eines Box-Inhalts (Text).	I	II
... kennen die unterschiedlichen Attribute von overflow.	I	II
... erlauben die Größenveränderung einer Box mit der CSS Eigenschaft resize.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Schattenverläufe für Boxen

CSS `box-shadow: [X] [Y] [blur] [spread] [color];`



- [X]** Horizontaler Versatz des Schattens.
 20px verschiebt den Schatten um 20 Pixel nach rechts.
 -10px verschiebt den Schatten um 10 Pixel nach links.
- [Y]** Vertikaler Versatz des Schattens
 20px verschiebt den Schatten um 20 Pixel nach unten.
 -15px verschiebt den Schatten um 15 Pixel nach oben.
- [blur]** Radius der Unschärfe (Weichzeicheneffekt).
 0px entspricht einer schärferen Darstellung.
- [spread]** Zusätzliche Vergrößerung des Schattens. Bei 0px hat der Schatten die gleiche Größe wie sein Element.
- [color]** Farbangabe

Weitere Attribute von box-shadow:

- inset** wirft den Schatten ins Innere der Box. **Inset** darf nur am Anfang oder am Ende der Attribute gesetzt werden. **ohne Wert**
- none** kein Schatten

```
.bilder {box-shadow: 10px 10px 5px 0px gray;}
div {box-shadow: inset -10px -10px 20px 10px black;}
```

Anzeige von Inhalten: overflow

Natürlich sollten HTML Elemente so gewählt sein, das der Inhalt problemlos dargestellt werden kann. Sollte es aber vorkommen, dass der Inhalt größer ist als die Eltern-Box, kann man mit der *overflow* Eigenschaft das Verhalten näher bestimmen.

CSS `overflow: [value];`



Für **value** stehen folgende Attribute zur Verfügung:

- auto** Browserabhängig
- scroll** Scrollbalken werden angezeigt
- visible** Inhalte sind sichtbar und ragen über das Elternelement hinaus.
- hidden** Inhalte werden an den Rändern unsichtbar (abgeschnitten).



Zusätzlich gibt es die CSS Eigenschaften *overflow-x* und *overflow-y* für horizontale und vertikale Inhalte die über die Box hinausragen.

```
div {overflow: hidden;}
```



Mit der CSS Eigenschaft *resize* erlaubt man dem/der Nutzer/in das Verändern der Größen eines Elements. Folgende Werte sind möglich: `both; horizontal; vertical; none;`

```
div {resize: both;}
```

Lernhandout 12.1 Formulare

Referenzcode: HCL121

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... wissen um den Nutzen von Formularen und Formularitems für die Verarbeitung mit Programmiersprachen oder Scriptsprachen bescheid.	I	I
... erstellen ein Formularelement (Parentelement, <form>).	I	II
... beschriften Formularelemente mit dem <label> Tag.	I	II
... verstehen das action-Attribut.	I	I
... verstehen das method-Attribut und unterscheiden zwischen get und post.	KI	II
... definieren einen Namen für das Formular.	I	II
... kennen noch weitere Attribute.	I	I
... verstehen und nutzen das <label> Element zur Beschriftung und kennen die Attribute form und for.	I	II
... nutzen <label> Elemente für das Barrierefreie Web. (Screenreader).	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Zahlreiche Anwendungen lassen sich mit Formularen in HTML verwirklichen (z. B. Mail-Formulare). Formulare bieten die Möglichkeit zur Eingabe von Informationen – die Verarbeitung dieser Informationen erfolgt dann durch eine Programmiersprache (z. B. Java), eine clientseitige Scriptsprache (z. B. JavaScript) oder einer serverseitigen-Sprache (z. B. PHP).

Die Formularelemente werden in einem `<form>` Tag plziert und mit einem `<label>` Tag beschriftet.

HTML `<form action="..." method="..." name="..."> ... </form>`



action	Im <code>action</code> Attribut wird eine URL eingetragen. In den meisten Fällen ist das die Adresse zu einem Script. <code>action="mailscript.php"</code> Wird das <code>action</code> Attribut weggelassen, dann werden die Daten an das aktuelle Dokument gesendet.
method	Legt die Methode fest, wie die Daten versendet werden sollen. Es gibt zwei mögliche Werte: <code>get</code> Die Daten werden sichtbar über die URL-Zeile des Browsers weitergegeben. <code>post</code> Die Übertragung erfolgt unsichtbar im Hintergrund.
name	Definiert einen Namen für das Formular. Dieser ist meist notwendig für die weitere Verarbeitung mit z. B. JavaScript.

Weitere Attribute:

target	Zielfenster
accept-charset	Angabe einer Zeichenkodierung
enctype	Angabe der Datenkodierung, Standardwert ist <code>text/plain</code>
autocomplete	Mit den Werten <code>on</code> oder <code>off</code> kann die Autovervollständigung des Browsers aktiviert, bzw. deaktiviert werden.
novalidate	Das Formular soll nicht auf Vollständigkeit überprüft werden.

`<form action="mailversenden.php" method="get" > ... </form>`

HTML `<label > ... </label>`



Das `<label>` Element hat zwei Vorteile. Erstens erleichtert es einen Screenreader das assoziierte Formelement vorzulesen. Zweitens wird bei einem Mouse-Click auf den Text innerhalb eines `<label>` Tags ein Focus auf das Formelement gelegt.

Um das `<label>` Element zu nutzen, muss im Formelement ein `id=""` gesetzt sein!

Attribute:

<code>form=""</code>	wird für das gesamte Formular verwendet.
<code>for=""</code>	bezeichnet ein Eingabeelement (muss identisch sein mit dem <code>id</code>).

```
<form action="senden.php" id="person" autocomplete="off" method="post">
  <label form="person">Namenseingabe</label>

  <label for="vorname">Vorname</label>
  <input type="text" name="vorname" id="vorname" maxlength="30">
  <button type="submit">Eingaben absenden</button>
</form>
```


Lernhandout 12.2 Textarea

Referenzcode: HCL122

Feinziele Die Schüler innen ...	Zielart	Taxonomie
... nutzen Textarea-Elemente für mehrzeilige Texteingaben.	I	II
... definieren die Anzahl der Zeilen und Zeichen mit den rows und cols Attributen.	I	II
... kennen und nutzen weitere Attribute es <textarea> Elements.	I	II
... stellen HTML Code innerhalb eines HTML Dokument mit dem <textarea> Elements dar.	I	II
... vergleichen <textarea> und <pre> für die Darstellung von HTML Code.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Mit dem `<textarea>` Element kann der Benutzer mehrzeilige Texteingaben tätigen.

HTML `<textarea name="" rows="" cols="">...</textarea>`



name	Eine Bezeichnung für die weitere serverseitige Verarbeitung.
rows	Anzahl der Zeilen (bestimmt damit auch die Höhe)
cols	Anzahl der Spalten bzw. die Anzahl der Zeichen pro Zeile

```
<form action="feedback.php" method="post" name="fb1" id="FB1">
  <label form="FB1">Rückmeldung</label>

  <label for="TF1">Haben Sie noch weitere Vorschläge?<br></label>
  <textarea name="tf1" id="TF1" rows="5" cols="50">
    Ihr Feedback...</textarea>
</form>
```



Weitere Attribute sind:

wrap	Bestimmt den Textumbruch. Mögliche Werte sind: <code>hard</code> oder <code>soft</code> .
readonly	Das Eingabefeld kann nur gelesen werden. Eine Eingabe ist also nicht möglich. <code>ohne Wert</code>
required	Eine Eingabe muss erfolgen um das Formular absenden zu können. <code>ohne Wert</code>
placeholder	Es kann ein Text bestimmt werden, welcher in der Textarea als Hinweis erscheint.
autofocus	Legt den Focus nach dem Laden der Seite auf das Textfeld. <code>ohne Wert</code>
maxlength	Maximale Länge der Eingabe (Anzahl der maximalen Buchstaben).

```
<textarea name="tear2" id="tear2"
  rows="4" cols="60" placeholder="Ihr Feedback ..."
  required autofocus maxlength="220" ></textarea>
```



Da das `<textarea>` Element kompromisslos alle Sonderzeichen, Leerzeichen und Zeilenschaltungen darstellt, kann es auch für die Darstellung von HTML Code verwendet werden.

```
<textarea rows="4" cols="100" readonly >
  <h1>Technisches Service</h1>
  <p>Hier finden Sie alle ...</p>
</textarea>
```



Das gleiche Ergebnis innerhalb eines `<pre>` Tags sieht dann so aus:

```
<pre>
  &lt;h1&gt;Technisches Service&lt;/h1&gt;
  &lt;p&gt;Hier finden Sie alle ...&lt;/p&gt;
</pre>
```

Lernhandout 12.3 Input

Referenzcode: HCL123

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... nutzen <input> Elemente für einzeilige Eingaben.	I	II
... bestimmen mit dem type Attribut die Art des <input> Elements.	I	II
... setzen einen eindeutigen Namen für das <input> Element, welcher für die spätere Verarbeitung in einer Script- oder Programmiersprache notwendig ist.	I	II
... beschriften <input> Elemente mit dem <label> Tag.	I	II
... kennen weitere generelle Attribute (size, readonly, step usw).	I	II
... kennen Attribute zur Validierung und Gestaltung.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Mit dem `<input>` Element können einzeilige Eingaben getätigt werden. Es ist sehr umfangreich und wird durch das `type` Attribut näher bestimmt. Das `<input>` Element ist ein Standalone-Tag und muss also nicht geschlossen werden.

HTML `<input type="" name="" value="">`



type	Beschreibt die Art des Eingabefelds. Als Standard wird <code>type="text"</code> angenommen.
name	Der Name des Eingabefelds wird für die spätere Verarbeitung in einer Script- oder Programmiersprache verwendet.
value	Ein vordefinierter Wert.

```
<form>
  <label for="Vorname" >Bitte geben Sie Ihren Vornamen ein</label><br>
  <input type="text" name="Vorname" id="Vorname" value="Kevin">
</form>
```



Weitere generelle Attribute

size	Definiert die Anzahl der Zeichen, die eingegeben werden können. z. B. bei <code>size="15"</code> hat das Feld eine Breite von 15 Zeichen.
readonly	Das Eingabefeld kann nur gelesen werden. Eine Eingabe ist nicht möglich. <code>ohne Wert</code>
min	Der Minimum-Wert bei numerischen oder Datumseingabefeldern. z. B. <code>min="200"</code> bedeutet eine Eingabe von mehr als 200.
max	Der Maximum-Wert bei numerischen oder Datumseingabefeldern. z. B. <code>max="1300"</code> bedeutet eine Eingabe von weniger als 1300.
step	Die Schrittweite der Eingabe bei numerischen oder bei Datumsfeldern. <code>step="any"</code> hebt die Schrittweite auf.
multiple	Erlaubt Mehrfacheingaben für Datei-Uploads oder Email-Eingaben. <code>ohne Wert</code>



Attribute zur Validierung und Gestaltung

minlength	Mindestlänge der Eingabe (in Zeichen) z. B. <code>minlength="5"</code> verlangt mindestens fünf Zeichen.
maxlength	Maximallänge der Eingabe (in Zeichen).
required	Pflichtfeld. Eine Eingabe muss erfolgen. <code>ohne Wert</code>
placeholder	Sichtbarer Hinweis im Inputfeld. Es verschwindet nach der Eingabe eines Zeichens.
autofocus	Nach dem Laden der Webseite, wird er Focus auf das Inputfeld gelegt. <code>ohne Wert</code>
autocomplete	Automatisches Vervollständigen von Formularfeldern durch den Browser. Als Werte sind <code>on</code> oder <code>off</code> möglich.

```
<input type="text" name="Nachname"
      size="20" maxlength="20" minlength="3"
      placeholder="Bitte Nachnamen eingeben!" required
      autocomplete="off" autofocus >
```

Übungsblatt 12.3 Input

Referenzcode: HCU123

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... recherchieren im Internet über den Unterschied zwischen get und post.	I	III
... stellen Überlegungen zur Verwendung von get und post an.	IK	III
... dokumentieren ihre Ergebnisse der Recherche und Überlegungen.	I	II
Übung B		
... scripten ein HTML Kommentar Formular.	I	II
... verwenden ein <form> Element mit den Attributen name, id, action, method.	I	II
... beschriften das <form> Element mit dem <label> Tag.	I	II
... fügen ein <input> Feld für einzeilige Texteingaben hinzu.	I	II
... setzen die notwendigen Attribute (z. B. placeholder) für das <input> Feld.	I	II
... fügen eine Textarea hinzu und definieren dafür die passenden Attribute (z. B. required).	I	II
... beschriften die Textarea mit dem <label> Tag.	I	II
... führen alle Notwendigen CSS Anpassungen durch (für Beschriftungen und Textfelder).	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

**Übung A: get und post**

- ☐ Recherchiere im Internet den Unterschied zwischen `get` und `post` als Methode für ein Formular.
- ☐ Wie funktioniert `get`? Wie funktioniert `post`?
- ☐ Welche Vor- und Nachteile haben `get` und `post`?
- ☐ Finde jeweils mind. drei Beispiele wo `get` bzw. `post` Sinn ergibt.

**Übung B: Kommentar-Formular**

- ☐ Erstelle ein neues HTML-Dokument mit einem Formular für Kommentare. Das Formular hat ein einzeliges Texteingabefeld für den Vor- und Nachnamen und ein mehrzeiliges Textfeld für den Kommentar.
- ☐ **<form> Eigenschaften:**
Das Formular wird im `comment.php` Script verarbeitet. Eine passende Methode ist selbst zu wählen. Beschrifte das Formular (`<label>`) mit **"Ihr Feedback"**
- ☐ **Vor- und Nachnamen (einzeilige Texteingabe)**
Die Attribute für den Namen und Größe sind selbst zu wählen. Es ist ein Pflichtfeld mit einem Platzhaltertext **"Hier bitte den Namen eingeben"**. Die Beschriftung lautet: **"Vor- und Nachname"**.
- ☐ **Kommentarfeld (mehrzeilige Texteingabe)**
Die Attribute für den Namen und Größe sind selbst zu wählen. Die maximale Länge der Eingabe ist auf 600 Zeichen beschränkt. Es ist ein Pflichtfeld. Die Beschriftung lautet: **"Kommentar"**.
- ☐ CSS Anpassung für die Beschriftungen:
Schriftart: Arial, Kursiv.
- ☐ CSS Anpassung für die Textfelder:
Schriftart: Courier New | Hintergrundfarbe: hellblau | Rahmen: 2 Pixel Dunkelgrau. | Wenn die Maus über das Textfeld fährt, soll sich die Hintergrundfarbe in weiß ändern

malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.

Ihr Feedback

Vor- und Nachname

Kommentar

Lernhandout 12.4 Input Typen

Referenzcode: HCL124

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... erstellen Eingabefelder für einzeiligen Text.	I	II
... erstellen Eingabefelder für Passworteingaben.	I	II
... erstellen Eingabefelder für eMail Adressen.	I	II
... erstellen Eingabefelder für Zahlen.	I	II
... kennen weitere Attribute für Zahleneingabefelder.	I	I
... erstellen Eingabefelder für Zahlen mit einem Schieberegler.	I	II
... kennen weitere Attribute für Eingabefelder mit Schieberegler.	I	I
... erstellen Eingabefelder für Telefonnummern.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Texteingabe

HTML `<input type="text">`

Passwörter

HTML `<input type="password">`



Im Eingabefeld werden Platzhalter anstatt der Zeichen angezeigt.
(Browserabhängig, meist aber Sterne *)

eMail-Adressen

HTML `<input type="email">`

Web-Adressen

HTML `<input type="url">`



Validiert die URL (eine vollständige Schreibweise ist notwendig: <http://www.css4.at>)
Mobile Geräte blenden z. B. .com automatisch in der Tastatur ein.



*Für die Typen text, password, email und url gibt es das Attribut inputmode.
Es hilft den mobilen Geräten die passende Tastatur zu wählen.
Werte sind z. B. verbatim (alphanumerisch), latin (Texteingabe mit Texteingabe-
hilfe) usw.*

Eingabe von Zahlen

HTML `<input type="number">`



Es können nur Zahlen eingegeben werden.
Weitere Attribute sind max, min, step und value.

Eingabe von Zahlen mit einem Schieberegler

HTML `<input type="range">`



Der Schieberegler (Slider) wird für die Eingabe von Zahlen verwendet.
Weitere Attribute sind max, min, step und value.

Mit dem orient Attribut kann man den Slider auch vertikal darstellen.
z. B. `<input type="range" orient="vertical">`

Telefonnummern

HTML `<input type="tel">`



Mobile Geräte blenden eine angepasste Tastatur ein.

Übungsblatt 12.4 Input Typen

Referenzcode: HCU124

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... erstellen ein typisches Anmeldeformular.	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
Übung B		
... erstellen ein Formular zur Berechnung von Sparzinsen.	I	II
... benutzen Eingabefelder für Zahlen und passe diese mit den notwendigen Attributen an.	I	II
... fügen ein Eingabefeld mit Schieberegler ein und passen dieses mit den notwendigen Attributen an.	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

**Übung A: Anmeldeformular**

- ☐ Erstelle ein typisches Anmeldeformular mit den Feldern eMail, Benutzername und Passwort.
- ☐ Beschrifte alle Felder mit dem `<label>` Element.
- ☐ Verschönere das Aussehen mit CSS. Freie kreative Übung.

eMail-Adresse
max.mustermann@johndoe.at

Benutzername
madmax96

Passwort
••••••••••••••••

**Übung B: Online Sparen**

- ☐ Erstelle ein Formular zur Berechnung von Sparzinsen für ein Online-Sparbuch.
- ☐ Zwei Eingabefelder
- ☐ Feld 1: Kapital
Nummernfeld, mindestens 1000 und maximal 50000. Schrittweite 500. Beschriftung durch ein `<label>` Element.
- ☐ Feld 2: Laufzeit in Jahren
Schieberegler, mindestens 1 und maximal 8, Schrittweite 1. Beschriftung durch ein `<label>` Element.
- ☐ Verändere das Aussehen mit CSS.

Kapital
15000

Laufzeit in Jahren
1 2 3 4 5 6 7 8

Lernhandout 12.5 Form Button

Referenzcode: HCL125

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... kennen die Funktionen von Buttons in einem Formular.	I	I
... verwenden <button> Elemente mit den Type Attributen button, submit und reset.	I	II
... kennen und nutzen das <button> Attribut: form.	I	II
... kennen und nutzen das <button> Attribut: formaction.	I	II
... kennen und nutzen das <button> Attribut: formmethod.	I	II
... kennen und nutzen das <button> Attribut: formenctype.	I	II
... kennen und nutzen das <button> Attribut: name.	I	II
... kennen und nutzen das <button> Attribut: value.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Für Formulare gibt es das `<button>` Element, welches anklickbare Schaltflächen erzeugt. Diese können ein Formular absenden, das Formular zurücksetzen oder eine clientseitige Aktion (z. B. für JavaScript) auslösen.

HTML `<button type="" >...</button>`



Zwischen dem Start und Endtag wird die Beschriftung des Buttons eingetragen. Das Aussehen kann selbstverständlich mit CSS angepasst werden. Für das `type` Attribut gibt es folgende Werte:

button	Auslösen einer clientseitigen Aktion
submit	Absenden eines Formulars, Standardwert
reset	Zurücksetzen des Formulars.

Passwort eingeben:

```
<form>
  <label for="eingabetext">Passwort eingeben:</label>
  <input type="password" name="eingabetext" id="eingabetext">

  <button type="button">Komplexität prüfen</button>
  <button type="reset">Zurücksetzen</button>
  <button type="submit">Absenden</button>
</form>
```



Weitere Attribute:

form	Wenn der Button außerhalb des Formulars ist, kann mit dem <code>form</code> Attribut der Name des Formulars angegeben werden.
formaction	Im Button kann die Aktion (URL) des Formulars festgelegt werden. Sollte es im <code><form></code> Element eine abweichende Methode geben, dann gilt die des <code><buttons></code>
formmethod	Im Button kann die Methode (Get, Post) definiert werden. Sollte es im <code><form></code> Element eine abweichende Methode geben, dann gilt die des <code><buttons></code>
formenctype	Codierung über den Button
name	Zur weiteren Verwendung in einem Script
value	Rückgabewert eines Buttons.

```
<form name="meinForm" id="meinForm">
  ...
</form>

<button form="meinForm" formaction="standard.php"
  formmethod="get" type="submit">Standard</button>

<button form="meinForm" formaction="premium.php"
  formmethod="post" type="submit">Premium</button>
```



Mit dem Attribut `formnovalidate` kann die Überprüfung von Pflichtfeldern (mit `require` gekennzeichnet) unterbunden werden. Es ist analog zum `novalidate` Attribut im `<form>` Element.

Übungsblatt 12.5 Formular Buttons

Referenzcode: HCU125

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... erstellen ein Formular nach Vorgaben.	I	II
... entscheiden sich für oder gegen ein Tabellen-Layout.	K	III
... reproduzieren ein CSS Layout nach einer bebilderten Vorlage.	I	II
... definieren Pflichtfelder.	I	II
... fügen Buttons für das Absenden und Zurücksetzen hinzu.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- 📄 Der Passierschein A38 ist aus dem Asterix-Abenteuer "Asterix erobert Rom" entnommen.

**Übung A: Passierschein A38**

- ☐ Erstelle ein Formular wie unten dargestellt.
- ☐ Wähle zwischen zwei Layoutlösungen für das Formular:
 - mit Tabelle (einfach aber mehr Schreibarbeit)
 - ohne Tabelle (knifflig aber dafür weniger Schreibarbeit)
- ☐ Versuche mit CSS das gleiche Erscheinungsbild wie unten zu erzeugen.
- ☐ Die Felder Vorname, Nachname, Telefon und eMail sind Pflichtfelder.
- ☐ Zwei Buttons zum Absenden und zum Zurücksetzen des Formulars.

Amtliches Antragsformular für den Passierschein A38

Alle mit einem * gekennzeichneten Feldern müssen ausgefüllt werden!

Titel:	<input type="text"/>	
Vorname:	<input type="text"/>	*
Nachname:	<input type="text"/>	*
Straße:	<input type="text"/>	
Postleitzahl:	<input type="text"/>	
Wohnort:	<input type="text"/>	
Telefon:	<input type="text"/>	*
eMail:	<input type="text"/>	*
Website:	<input type="text"/>	
<input type="button" value="Passierschein A38 beantragen"/>		<input type="button" value="Zurücksetzen"/>

Lernhandout 12.6 Dialogeingaben

Referenzcode: HCL126

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... erstellen ein Input-Feld für einen Datei-Upload.	I	II
... kennen und nutzen die Attribute für Datei-Upload-Dialoge.	I	II
... erstellen ein Input-Feld für die Farbauswahl.	I	II
... erstellen ein Input-Feld für Datum und Uhrzeit.	I	II
... erstellen ein Input-Feld für versteckte Elemente.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- 📄 Die Dialoge können von Browser zu Browser unterschiedlich dargestellt werden.

Datei-Upload

HTML `<input type="file" >`


Der Datei-Upload funktioniert nur mit der `Post`-Methode (im `<form>` oder im `<button>`). Zusätzlich muss im `<form>` Element noch das Attribut `enctype="multipart/form-data"` hinzugefügt werden, denn sonst wird nur das Name der Datei übertragen. Weitere Attribute:

accept Definiert die möglichen Dateitypen (MIME-Typen) für den Upload.
 z. B. nur Textdateien: `<input type="file" accept="text/*">`
 nur JPG Bilder: `<input type="file" accept="image/jpeg">`

multiple Mehrfachauswahl **ohne Wert**

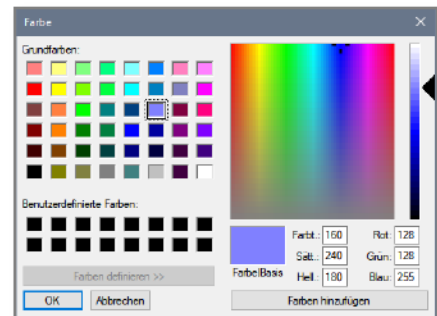
```
<form action="upload.php" enctype="multipart/form-data" method="post">
  <label for="datei" >Bilder hochladen</label>
  <input type="file" accept="image/*" name="datei" id="datei" multiple>
</form>
```

Farbauswahl

HTML `<input type="color" >`


Der Rückgabewert der ausgewählten Farbe ist ein HEX-Wert.

```
<form method="get">
  <label for="farbe" >Lieblingsfarbe</label>
  <input type="color" value="#ffffff"
    id="farbe" name="farbe" >
</form>
```



Datum und Uhrzeit

HTML `<input type="date" >`


Mit dem `type="date"` Attribut werden Datumseingaben abgefragt. In der Regel stellt der Browser ein Datumsfeld oder ein Kalender-Dialog zur Verfügung.

Artverwandt sind:

Wochenangaben: `<input type="week" >`
 Monatsangaben: `<input type="month" >`
 Uhrzeiten: `<input type="time" >`
 Kombination: `<input type="datetime-local" >`

Versteckte Elemente

HTML `<input type="hidden" value="" >`


Das Eingabefeld ist für den Benutzer unsichtbar. Der Wert wird aber beim Absenden übergeben.

Übungsblatt 12.6 Dialoge

Referenzcode: HCU126

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... fügen ein Eingabefeld für eine Farbwahl in ein HTML Dokument ein.	I	II
... beschriften das Eingabefeld.	I	II
... testen das Eingabefeld mit beigelegter (Report_GET.html) JavaScript Applikation.	I	II
Übung B		
... erstellen ein Formular für eine Online-Bewerbung.	I	II
... finden die passenden Input-Felder für ein klassisches Online-Bewerbungsformular.	KI	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- Die Report_GET.html Datei übernimmt Formulardaten und wertet diese mit JavaScript aus. Also wird für das action-Attribut: Report_Get.html und für die Methode: GET angegeben.



Übung A: Farbauswahl

- ☐ Erstelle ein neues HTML Dokument.
- ☐ Füge ein Eingabefeld für Farben hinzu.
- ☐ Beschrifte das Eingabefeld mit "**Farbe der Mannschaft**". `<label>`
- ☐ Verwende als Methode `GET` mit dem `Report_GET.html` über den Submit-Button und nicht als Attribut im `<form>` Element.



Übung B: Online Bewerbung

- ☐ Erstelle ein neues HTML Dokument mit einem Formular zur Online Bewerbung.
- ☐ Folgende Eingabefelder sollten eingebunden sein:
Vor- und Nachname, E-Mail-Adresse, Geburtsdatum, Dateiapload für den Lebenslauf (nur PDF) und ein mehrzeiliges Textfeld für das Motivationsschreiben.
- ☐ Gestalte das Formular mit CSS anspruchsvoll. Freie kreative Übung.

The screenshot shows a web browser window with the title 'Online Bewerbung'. The browser's address bar contains the text 'Mit Google suchen oder Adresse eing'. The page content includes a heading 'Online Bewerbung', a subheading 'Ihre Angaben', and several form fields: 'Vor- und Nachname' (filled with 'Max Mustermann'), 'eMail-Adresse' (filled with 'max.mustermann@gmail.com'), 'Geburtsdatum' (filled with 'TT . MM . JJJJ'), 'Lebenslauf hochladen (nur PDF)' (with a 'Durchsuchen...' button and the text 'Keine Datei ausgewählt.'), and 'Motivationsschreiben' (a large text area). At the bottom, there is a button labeled 'Bewerbung absenden'.

Lernhandout 12.7 Auswahleingaben

Referenzcode: HCL127

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... binden Radio-Buttons in ein HTML Dokument ein.	I	II
... kennen die Attribute von Radio-Buttons.	I	I
... binden Checkbox Felder (Kontrollfelder) in ein HTML Dokument ein.	I	II
... kennen die Attribute von Checkbox-Feldern.	I	I
... stellen Überlegungen zu Radio-Buttons und Checkbox-Feldern an und analysieren, wann die jeweilige Verwendung Sinn ergibt.	IK	III
... definieren ein <fieldset> für Formularelemente.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Radio-Buttons

HTML `<input type="radio" name="" value="">`


Radio-Buttons sind eine Gruppe von Auswahlbuttons in der nur eine Eingabe erlaubt ist. Wird also ein Button angeklickt, wird ein anderer abgewählt. Folgende Attribute sind möglich:

- name** Alle Radio-Buttons mit dem gleichen `name` Attributwert, gehören zu einer Gruppe.
- value** Definiert den Wert, der beim Absenden des Formulars übergeben wird.
- checked** Ein bestimmter Radio-Button wird vorselektiert. ohne Wert
- disabled** Zum Deaktivieren. Es wird ausgegraut dargestellt. ohne Wert

```
<form>
  <input type="radio" name="Zahlung" value="Nachnahme" id="nn">
  <label for="nn"> per Nachnahme</label>
  <input type="radio" name="Zahlung" value="Vorauskassa" id="vk">
  <label for="vk"> per Vorauskassa</label>
</form>
```

Checkbox

HTML `<input type="checkbox" name="">`


Eine Checkbox ist ein anklickbares Kontrollfeld. Per Klick erscheint in der Regel ein Häkchen bei den ausgewählten Feldern.

- name** Jedes Checkbox Element sollte einen eigenen Namen bekommen.
- checked** Zum Vorselektieren von Checkboxes. ohne Wert
- disabled** Zum Deaktivieren. Es wird ausgegraut dargestellt. ohne Wert

```
<form method="get">
  <input type="checkbox" name="abo1" id="gabo">
  <label for="gabo"> Gratis Ausgabe bestellen</label><br>

  <input type="checkbox" name="newsletter" id="nwsl" checked>
  <label for="nwsl"> Newsletter bestellen</label>

  <button type="submit">Absenden</button>
</form>
```

HTML `<fieldset> <legend> ... </legend> ... </fieldset>`


Mit dem `<fieldset>` Element kann man Formelemente innerhalb eines Formulars gruppieren. Es sind die Attribute `name`, `disabled` und `form` verfügbar. Das `<legend>` Element wird zur Beschriftung des `<fieldset>` verwendet.

```
<fieldset name="Zahlgruppe">
  <legend> Zahlungsarten</legend>
  ....
</fieldset>
```

Übungsblatt 12.7 Auswahleingabe

Referenzcode: HCU127

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... erstellen ein Bestellformular für einen Lieferservice.	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
... fügen Auswahleingaben (Radio, Checkboxes) hinzu.	I	II
... beschriften alle Felder und fügen ein <fieldset> hinzu.	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
... testen das HTML Dokument mit der Report_GET.html	I	II
Übung B		
... erweitern eine bereits gelöste und gespeicherte Arbeit (Dateimanagement).	S	II
... optimieren das HTML Dokument für die Darstellung auf Smartphones.	I	II
... testen die Arbeit auf einem Smartphone.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen



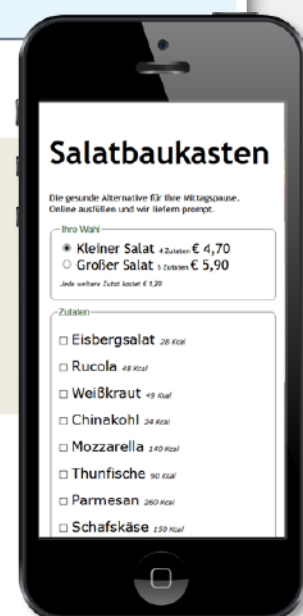
Übung A: Salat Online

- ☐ Erstelle ein Bestellformular für einen Salat Lieferbetrieb.
- ☐ Auswahl zwischen einem großen und einem kleinen Salat.
- ☐ Mindestens 12 Zutaten mit Kalorienangaben (Mehrfachauswahl).
- ☐ Mindestens 4 Dressings mit Kalorienangaben (nur eine Auswahl möglich).
- ☐ Verwende `<fieldset>` Elemente zur optischen Trennung.
- ☐ Beschrifte alle Felder mit `<label>`
- ☐ Gestalte die Seite mit CSS – freie kreative Übung.
- ☐ Teste die Seite mit der `Report_GET.html`



Übung B: Salat Online Mobil

- ☐ Öffne deine Lösung von Übung 12.7 A: Salat Online.
- ☐ Optimierte die Seite für mobile Geräte (Smartphone, Tablet usw.)
- ☐ Teste die Seite auf einem Smartphone!



Lernhandout 12.8 Auswahllisten

Referenzcode: HCL128

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... erstellen Auswahllisten (Drop-Down-Listen) mit dem <select> Element.	I	II
... erstellen Auswahllistenpunkte mit dem <option> Tag.	I	II
... kennen die Attribute von <select> und <option>.	I	I
... begrenzen Auswahllisten durch die Anzahl der dargestellten Zeilen.	I	II
... selektieren einen Auswahlpunkt vor.	I	II
... stellen Überlegungen zur Mehrfachauswahl (multiple) an.	K	III

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Drop-Down-Liste

HTML `<select name="" ><option>...</option></select>`


Das `<select>` Element stellt eine Auswahlliste, bzw. ein Drop-Down-Feld zur Verfügung. Innerhalb des `<select>` Elements werden über den `<option>` Tag die Einträge festgelegt.

Attribute für das `<select>` Element.

<code>name</code>	Name des Elements (Variablenname).
<code>size</code>	Anzahl der sichtbaren Zeilen. Ohne Size Attribut wird ein Drop-Down-Feld angezeigt.
<code>multiple</code>	Ermöglicht eine Mehrfachauswahl ohne Wert
<code>required</code>	Die Eingabe ist erforderlich (Pflichtfeld) ohne Wert

Attribute für das `<option>` Element.

<code>value</code>	Übergabewert, wenn es angeklickt wird.
<code>selected</code>	Eine Option wird vorausgewählt. ohne Wert

```
<form action="Report_GET.html" method="get">

  <label for="philos">Berühmte Philosophen:</label><br>

  <select name="philos" id="philos">
    <option>Friedrich Nietzsche</option>
    <option>Ludwig Wittgenstein</option>
    <option>Immanuel Kant</option>
    <option>David Hume</option>
  </select>

  <button type="submit">Absenden</button>

</form>
```

Berühmte Philosophen:

Immanuel Kant	Absenden
Friedrich Nietzsche	
Ludwig Wittgenstein	
Immanuel Kant	
David Hume	



Hier eine Auswahlliste mit drei Zeilen und `value` Attribute für das `<option>` Element. Das Auswahlfeld `value="2typ"` wurde vorselektiert.

```
<select name="philos" id="philos" size="3">
  <option value="1typ">Friedrich Nietzsche</option>
  <option value="2typ" selected>Ludwig Wittgenstein</option>
  <option value="3typ">Immanuel Kant</option>
  <option value="4typ">David Hume</option>
</select>
```

Berühmte Philosophen:

Friedrich Nietzsche	Absenden
Ludwig Wittgenstein	
Immanuel Kant	



Eine Mehrfachauswahl (mit dem `multiple` Attribut) ist für den Anwender nicht immer klar ersichtlich (z. B. mit gedrückter STRG-Taste selektieren). Checkboxes eignen sich besser für eine Mehrfachauswahl.

Übungsblatt 12.8 Auswahllisten

Referenzcode: HCU128

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... erstellen eine HTML Seite nach einer bebilderten Vorlage.	I	II
... fügen vier Auswahllisten hinzu.	I	II
... nutzen <fieldset> Elemente zur Beschriftung.	I	II
... fügen ein Drop-Down-Feld hinzu.	I	II
Übung B		
... erstellen eine Webseite für ein Gewinnspiel.	I	II
... fügen Input-Felder und Auswahllisten getreu der Vorgaben hinzu.	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

- 📄 Mit clever gesetzten <label> Beschriftungen kann ein Klick auf eine Antwort das Auswahlfeld aktivieren.
- 📄 Es gehört zum guten Ton im Web Newsletter Abonnements nicht vorzuselektieren.

**Übung A: Softwareliste**

- ☐ Erstelle ein Formular wie unten dargestellt.
- ☐ Vier Auswahllisten.
- ☐ Fieldsets als Beschriftung
- ☐ Ein Drop-Down-Feld (Ausgabe: HTML, XML, Plain Text).

Betriebssystem	Browser	Suchmaschine	Coding
Windows	Edge	Google	C#
OSx	Firefox	Yahoo	Visual Basic
Linux	Chrome	Web.de	Java
Unix	Safari	Bing	Pascal
Anderes	Anderer	Anderer	Anderer

Ausgabe:

**Übung B: Gewinnspiel**

- ☐ Erstelle eine Webseite für ein Gewinnspiel
- ☐ Mindestens drei Radio-Buttons
- ☐ Drei Textfelder für den Namen, eMail-Adresse und Telefon.
- ☐ Ein Drop-Down-Feld
- ☐ Mindestens zwei Checkboxes.
- ☐ Freie kreative Übung.

GEWINNSPIEL

Beantworten Sie eine einfache Frage und gewinnen Sie ein Wochenende in Salzburg in einem 5 Sternehotel und zwei Tickets für die Festspiele!

In welchem Jahr wurde Mozart geboren?

☐ Antwort 1: 1721
☐ Antwort 2: 1756
☐ Antwort 3: 1803

Vor und Nachname
E-Mail-Adresse
Telefonnummer

Bundesland
Wien

☒ Ja, ich möchte am Gewinnspiel teilnehmen!
☐ Ja, ich möchte den Newsletter abonnieren!

Lernhandout 13.1 Animationen

Referenzcode: HCL131

Feinziele Die Schüler innen ...	Zielart	Taxonomie
... verstehen die Idee der Animation in CSS.	I	III
... definieren @keyframes Regeln mit Start- und Endeigenschaften.	I	II
... deklarieren die Animation mit der animation-name Eigenschaft.	I	II
... verändern die Dauer der Animation mit der animation-duration Eigenschaft.	I	II
... verwandeln <div> Quadrate in Kreise mittels einer Animation.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Mit CSS ist es möglich, Animationen ganz ohne Flash oder JavaScript zu erstellen. Dabei werden die Start-Eigenschaften eines Elements bis zu den End-Eigenschaften verändert. Der Übergang verläuft flüssig. Für die Animation wird zuerst eine `@keyframe` Regel erstellt.

CSS `@keyframes meineAnimation{...}`



Hier in diesem Beispiel ist `meineAnimation` der Name der Animation. Natürlich kann auch ein anderer Name gewählt werden. Innerhalb der `@keyframes` Regel wird die Start-Eigenschaft mit `from {...}` und die End-Eigenschaften mit `to {...}` festgelegt.

```
@keyframes meineAnimation {
  from {background-color:blue;}
  to   {background-color:red;}
}
```

Nachdem die Animation über die `@keyframes` Regel erstellt wurde, muss sie über einen Selektor (ID, Class, usw) dem Element zugewiesen werden. Dafür wird mit `animation-name` der Name der Animation angegeben und mit `animation-duration` die Dauer der Animation festgelegt. Nach der Animation nimmt das Element wieder seinen originalen Style an.

CSS `animation-name: [Name];`



Für [Name] wird der Namen der `@keyframes` Regel eingetragen. In diesem Beispiel: `animation-name: meineAnimation;`

CSS `animation-duration: [Value];`



Für [Value] wird die Dauer der Animation in Sekunden eingetragen. In diesem Beispiel vier Sekunden: `animation-duration: 4s;`

```
div {width: 500px; height: 500px;
background-color:red;
animation-name: meineAnimation;
animation-duration: 4s;}
```



Das Beispiel verwandelt alle `<div>` Elemente vom Quadrat in einen Kreis. Dauer der Animation ist 5 Sekunden.

```
@keyframes rundeEcken {
  from {border-radius:0px;}
  to   {border-radius: 500px;}
}

div {width: 500px; height: 500px; margin:auto;
border: 20px solid red; border-radius: 500px;
animation-name: rundeEcken;
animation-duration: 5s;}
```

Lernhandout 13.2 Animationen

Referenzcode: HCL132

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... definieren Animationsschritte über Prozentangaben in der @keyframes Regel.	I	II
... verzögern den Animationsstart mit der CSS Eigenschaft animation-delay.	I	II
... definieren Wiederholungen von Animationen mit der CSS Eigenschaft animation-iteration-count.	I	II
... nutzen die CSS Eigenschaft animation-play-state um den Zustand einer Animation festzulegen.	I	II

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

Die `@keyframes` Regel ist nicht nur auf ein `form` und `to` beschränkt. Die einzelnen Animationschritte können auch mit Prozentangaben definiert werden. (von 0% bis 100%).

CSS `@keyframes animationsName {0-100%{...}}`

```
@keyframes bewegung {
  0% {top: 0px; left: 0px; background: red;}
  25% {top: 0px; left: 200px; background: blue;}
  50% {top: 200px; left: 200px; background: yellow;}
  75% {top: 200px; left: 0px; background: green;}
  100% {top: 0px; left: 0px; background: red;}
}
```

Verzögerung

CSS `animation-delay: ;`



Die Eigenschaft bestimmt die Verzögerung des Animationsstarts.
`animation-delay: 3s;` wartet drei Sekunden bis die Animation startet.

```
#dasDIV {width:200px; height:200px; background:green; position:fixed;
  animation-name:bewegung; animation-duration:3s;
  animation-delay:5s;}
~~~~~
<div id="dasDIV">Warte 5 Sekunden</div>
```

Wiederholungen

CSS `animation-iteration-count: ;`



Die Eigenschaft bestimmt die Wiederholungen der Animation.
`animation-iteration-count: 4;` wiederholt die Animation viermal.
`animation-iteration-count: infinite;` wiederholt ohne Ende.

```
#dasDIV {width:200px; height:200px; background:green; position:fixed;
  animation-name:bewegung; animation-duration:3s;
  animation-iteration-count:3;}
~~~~~
<div id="dasDIV">Drei Wiederholungen</div>
```

Zustand

CSS `animation-play-state: ;`



Bestimmt ob eine Animation pausiert oder läuft.
`animation-play-state: paused;` Die Animation pausiert.
`animation-play-state: running;` Die Animation läuft (Standardwert)

```
#dasDIV {width:200px; height:200px; background:green; position:fixed;
  animation-name:bewegung; animation-duration:3s;
  animation-iteration-count:infinite;}

#dasDIV:hover {animation-play-state:paused;}
~~~~~
<div id="dasDIV">Stoppe mich mit der Maus</div>
```


Lernhandout 13.3 Animationen

Referenzcode: HCL133

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... wechseln bei jedem Animationsdurchgang seine Richtung mit der CSS Eigenschaft animation-direction	I	II
... definieren mit der CSS Eigenschaft animation-fill-mode den Zustand nach Beendigung einer Animation.	I	II
... legen mit einer Timinigfunktion die Dauer zwischen den Schlüsselbildern fest.	I	II
... fassen mehrere Animationseigenschaften in der CSS Eigenschaft animation zusammen.	I	II
... kennen den Browsersupport für Animationen mit CSS.	I	I

Zielarten: A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene
Taxonomien: I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

CSS `animation-direction: [value];`

[value] Wechselt bei jedem Durchgang (Zyklus) seine Richtung.

- `normal` Die Animation wird immer vorwärts abgespielt.
- `alternate` Die Animation wechselt nach jedem Durchlauf die Richtung.
- `reverse` Die Animation wird immer rückwärts abgespielt.

CSS `animation-fill-mode: [value];`

[value] Definiert den Zustand nach der Animation.

- `forwards` Das Element übernimmt die Werte des letzten Animationsschrittes.
- `backwards` Das Element übernimmt die Werte des ersten Animationsschrittes
- `both` Verbindet die Werte von `forwards` und `backwards`.

CSS `animation-timing-function: [value];`

[value] Die Timingfunktion bezieht sich auf die Dauer zwischen den Schlüsselbildern. .

`ease`, `ease-in`, `ease-out`, `ease-in-out`, `linear`, `step-start`, `step-end`, `cubic-bezier()`, `steps()` usw.

CSS `animation: ;`

Die einfache CSS Eigenschaft `animation` fasst alle anderen animation-Eigenschaften zusammen.








Das Beispiel lässt ein Element aus der Bildschirmmitte hineinfliegen!

```
#meinDIV {position:fixed; background-color:#CCC;
  animation:ausZentrum 3s;
  animation-fill-mode: forwards;}

@keyframes ausZentrum {
  from {height:1%; width:1%; top:48%; right:48%;}
  to   {height:80%; width:80%; top:10%; right:10%;}
}
```

~~~~~  
<div id="meinDIV">Hinweis</div>

| Browsersupport          |  |  |  |  |  |
|-------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| @keyframes<br>animation | 43.0                                                                                | 10.0                                                                                | 16                                                                                  | 9.0                                                                                   | 30.0                                                                                  |

## Übungsblatt 13.3 Animationen

Referenzcode: HCU133

| Feinziele                                                                                                | Zielart | Taxonomie |
|----------------------------------------------------------------------------------------------------------|---------|-----------|
| Die Schüler innen ...                                                                                    |         |           |
| <b>Übung A</b>                                                                                           |         |           |
| ... erstellen ein animiertes Balkendiagramm.                                                             | I       | II        |
| ... animieren die Balken von links nach rechts.                                                          | I       | II        |
| ... definieren eine Animationsverzögerung.                                                               | I       | II        |
| ... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit. | A       | II        |
| <b>Übung B</b>                                                                                           |         |           |
| ... ordnen vier Rechtecke (div) im Zentrum des Viewports an.                                             | I       | II        |
| ... weisen den Rechtecken (div) unterschiedliche CSS Eigenschaften zu (z. B. Farbe, usw.)                | I       | II        |
| ... animieren die Rechtecke (div) so, dass sie von der Ecke ins Zentrum des Viewports 'fliegen'.         | I       | II        |
| ... verändern die Größe der Rechtecke (div) mit CSS Animationen.                                         | I       | II        |
| ... weisen den Rechtecken (div) unterschiedliche Timing-Funktionen zu.                                   | I       | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

---



### Übung A: Abstimmungsergebnis

- ☐ Erstelle ein animiertes Balkendiagramm.
- ☐ Die Balken (Ja 45%, Nein 37%, Egal 18%) haben eine unterschiedliche Farbe
- ☐ Die Balken wachsen von links nach rechts.
- ☐ Die Balken erscheinen hintereinander (Verzögerung).
- ☐ Verschönere das Diagramm mit CSS.

## Abstimmungsergebnis

Ja 45%

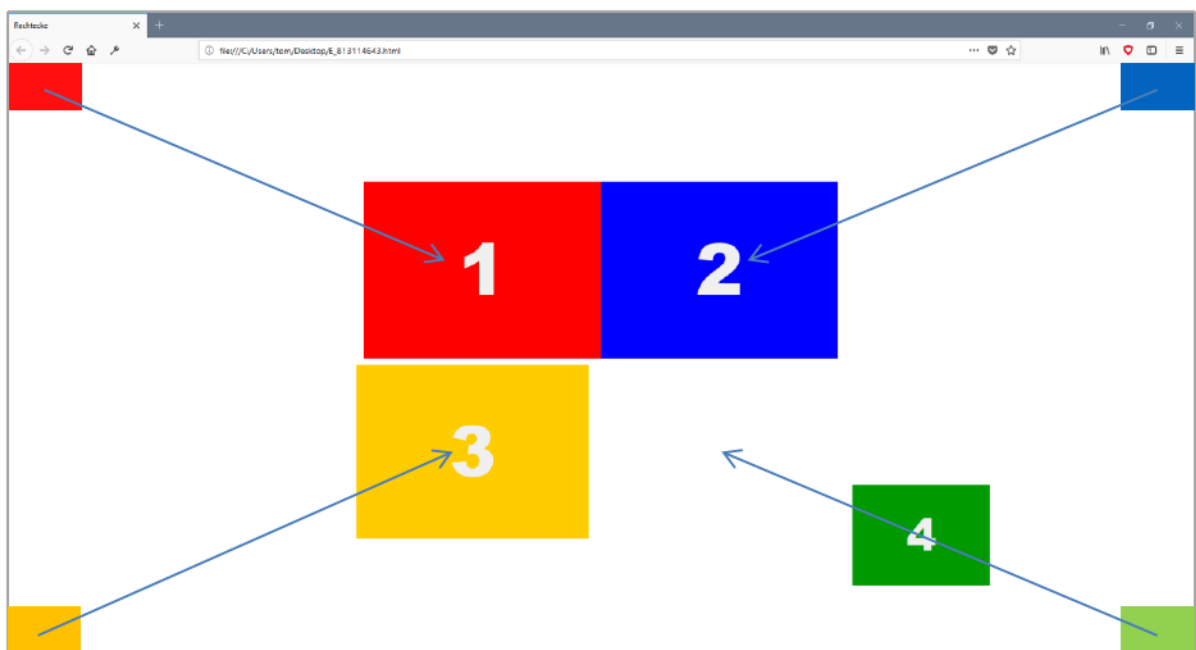
Nein 37%

Egal 18%



### Übung B: Rechtecke ins Zentrum

- ☐ Vier rechteckige Div's sollen im Zentrum angeordnet werden.
- ☐ Die Div's dürfen sich nicht überlappen.
- ☐ Jedes Div hat eine eigene unterschiedliche Farbe.
- ☐ Die Div' fliegen von der Ecke ins Zentrum
  - 1: von links oben
  - 2: von rechts oben
  - 3: von links unten
  - 4: von rechts unten
- ☐ Die Div's sollen auch in Höhe und Breite animiert werden (von Klein nach Groß).
- ☐ Jedes Div hat eine unterschiedliche Timing-Funktion.



## Lernhandout 13.4 Transformieren

Referenzcode: HCL134

| Feinziele                                                                                         | Zielart | Taxonomie |
|---------------------------------------------------------------------------------------------------|---------|-----------|
| Die Schüler innen ...                                                                             |         |           |
| ... können mit der CSS Eigenschaft transform 2D und 3D Veränderungen an HTML Elementen vornehmen. | I       | II        |
| ... skalieren Elemente mit transform: scale(x,y).                                                 | I       | II        |
| ... erstellen ein Beispiel mit einer Unterstreichung von der Mitte heraus.                        | I       | II        |
| ... drehen Elemente mit transform: rotate(winkel).                                                | I       | II        |
| ... kennen die Winkelmaße grad, rad, turn.                                                        | I       | I         |
| ... simulieren einen analogen Sekundenzeiger einer Uhr.                                           | I       | II        |

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

Zum Beispiel mit der animierten Unterstreichung: Sollte die Unterstreichung zu wenig abstrakt und zu besonders sein, so kann das Beispiel ohne das Pseudoelement ::before vorgezeigt werden:

```
.mitUL {font-size:3em; position:relative; display:inline-block;
        animation: meineAniStart 2s;}

@keyframes meineAniStart {
    from {transform: scaleX(0);
    to   {transform: scaleX(1);} }
~~~~~
<h1 class="mitUL">Unsere Abteilung als Profit-Center</h1>
```

Mit der CSS Eigenschaft `transform` lassen sich 2D und 3D Veränderungen bewerkstelligen. Wir werden uns die 2D Veränderungen ansehen, weil damit die meisten 3D Veränderungen auch visualisiert werden können.

## Skalieren

**CSS** `transform: scale(x,y);`



Verändert die Größe eines Elements über die x-Achse (Breite) und die y-Achse (Höhe). `scale(1,1)` ist die Originalgröße, `scale(0.5,0.5)` ist die halbe Größe und `scale(2,2)` vergrößert das Element um das doppelte.

Die x und y Achse lässt sich auch direkt ansprechen:

`transform: scaleX(2); transform: scaleY(1.4);`



Für die `scale(x,y)` Funktion sind positive und negative Zahlen (Dezimal und Ganzzahlen) möglich. Bei `transform: scale(1,-1)` wird das Element gespiegelt.



Das Beispiel zeigt eine animierte Unterstreichung von der Mitte heraus.

```
.mitUL {font-size:3em; position:relative; display:inline-block;}

.mitUL:before {content: ""; position: absolute; width: 100%;
 border-bottom: 3px solid black; bottom: 0; left: 0;
 animation: meineAniStart 2s;}

@keyframes meineAniStart {
 from {transform: scaleX(0);}
 to {transform: scaleX(1);} }
~~~~~
<h1 class="mitUL">Unsere Abteilung als Profit-Center</h1>
```

## Drehen

**CSS** `transform: rotate(winkel);`



Dreht ein Element im Uhrzeigersinn. Bei einem negativen Winkel gegen den Uhrzeigersinn.

Der Winkel-Wert wird als Zahl plus `deg` angegeben:

`transform: rotate(45deg);`

Andere Winkelmaße sind `grad` (Gon), `rad` (Radiant), `turn` (Vollwinkel).

Bei `deg` entspricht eine Kreismrundung 360°.



Das Beispiel simuliert einen analogen Sekundenzeiger.

```
#drehe {animation: meineDrehung infinite 60s linear ;}

@keyframes meineDrehung {
    from {transform: rotate(-90deg);}
    to   {transform: rotate(270deg);} }
~~~~~
<div id="drehe">Sekunden ⇒div>
```



# Übungsblatt 13.4 Transformieren

Referenzcode: HCU134

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... fügen drei Spielkarten in ein HTML Dokument ein.	I	II
... setzen die Pseudoklasse :target um ein aufdecken zu bewerkstelligen.	I	II
... animieren das Umdrehen einer Karte mit transform:scale.	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
Übung B		
... suchen im Internet nach Bildern von berühmten Malern.	A	I
... erstellen aus dem Bild eine Polaroid-Optik.	I	II
... fügen diverse CSS Eigenschaften dem Bild hinzu.	I	II
... drehen das Polaroid mit transform.	I	II
... beschriften das Polaroid mit dem Namen des Künstlers.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

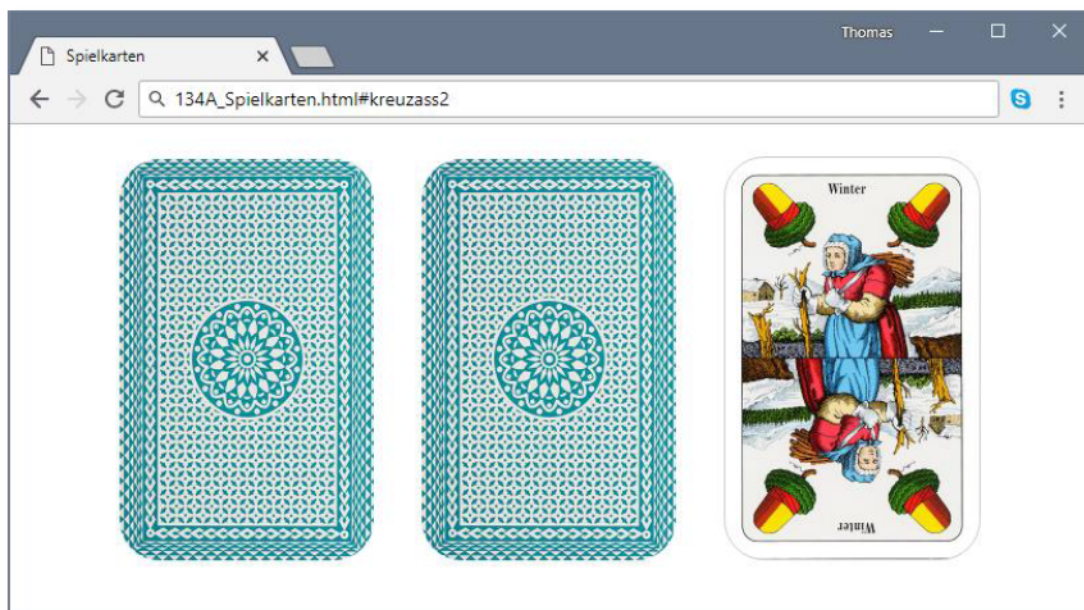
Anmerkungen

---



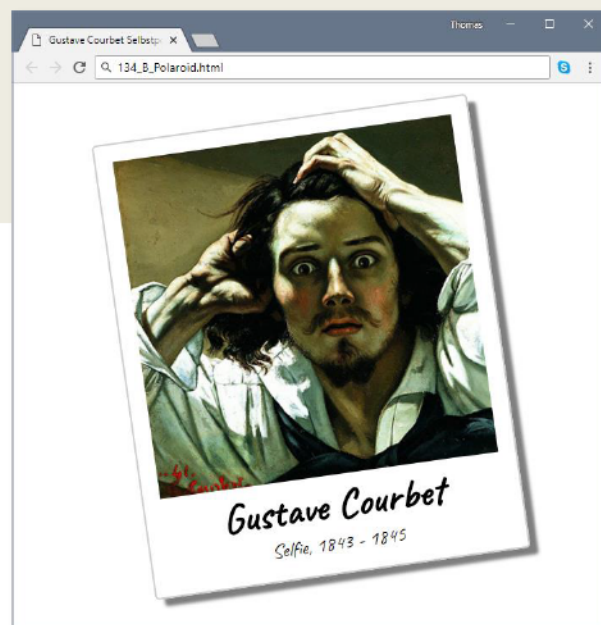
## Übung A: Spielkarten

- ☐ Erstelle eine Webseite mit drei Spielkarten. (Bilder dazu sind sicher im Internet zu finden).
- ☐ Die Spielkarten sind zum Start verdeckt aufgelegt.
- ☐ Durch einen Klick auf eine Karte wird diese umgedreht.
- ☐ Es darf immer nur eine Spielkarte aufgedeckt sein.
- ☐ Das Umdrehen soll animiert sein.
- ☐ Freie kreative Übung.



## Übung B: Polaroid

- ☐ Suche im Internet nach einem Selbstportrait eines berühmten Malers.
- ☐ Erstelle eine Webseite mit dem Bild in einer Polaroid Optik.
- ☐ Füge einen Schatten hinzu.
- ☐ Drehe das Polaroid mit CSS.
- ☐ Beschrifte es mit dem Namen des Künstlers.



## Lernhandout 13.5 Transformieren

Referenzcode: HCL135

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... verzerren ein Element über einen Winkel mit transform: skew(x,y).	I	II
... verschieben Elemente mit transform: translate(x,y).	I	II
... können einen einfachen Tab-Reiter erstellen.	I	II
... wissen, dass sich komplexere Verformungen mit transform: matrix möglich sind.	I	I

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

---

## Verzerren

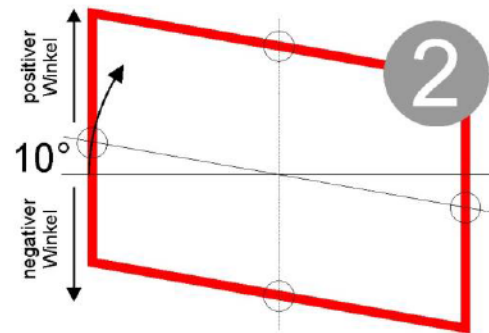
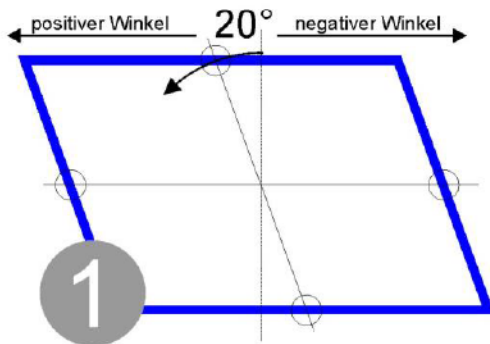
**CSS** `transform: skew(x,y);`


Verzerrt ein Element über einen Winkel für die x oder y Achse.

Die x und y Achse lässt sich auch direkt ansprechen:

Form 1: `transform: skewX(20deg);`

Form 2: `transform: skewY(10deg);`



## Verschieben

**CSS** `transform: translate(x,y);`


Verschiebt ein Element über die X-Achse und Y-Achse.

Als Werte können Pixel, cm, %, em usw. gesetzt werden. Negative Werte sind ebenfalls möglich.

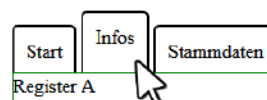
Die Eigenschaft kann auch direkt angesprochen werden:

`transform: translateX(2cm);` Zwei Zentimeter nach rechts

`transform: translateY(-35px);` 35 Pixel nach oben



Ein einfacher Tab-Reiter mit  
`transform: translate`



```
.register {height:2em; display:inline-block;
border-radius:5px 5px 0px 0px; padding:10px;
border: 2px solid black;}

.register:hover {transform: translateY(-0.7em);}

#inhalt {background-color:#FFF; border:1px solid green; height:10em;
transform:translateY(-1em);}
~~~~~
<nav><span class="register">Start</span>
<span class="register">Infos</span>
<span class="register">Stammdaten</span></nav>
<div id="inhalt">Register A</div>
```



Komplexere Verformungen von Elementen lassen sich mit der `matrix()` Funktion realisieren. z. B. `transform: matrix(1, 2, -1, 1, 80, 80);`

## Lernhandout 13.6 Transitions

Referenzcode: HCL136

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... definieren Übergänge ohne Animationen oder Keyframe Regeln.	I	II
... begreifen den Unterschied zwischen Übergängen und Animationen.	I	I
... bestimmen eine besondere CSS Eigenschaft für den Übergang mit transition-property.	I	II
... definieren die Dauer des Überganges mit transitions-duration.	I	II
... ergänzen einen Übergang mit einer transition-timing-function (Geschwindigkeit des Überganges).	I	II
... verzögern den Start des Überganges (Wartezeit) mit transition-delay.	I	II
... kennen und nutzen die Shorthand-Eigenschaft transition.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

---

*Auch ohne Animationen oder Keyframe Regeln, lassen sich Übergänge mit CSS definieren. Die `transitions` Eigenschaft kommt immer dann zu tragen, wenn sich ein Element verändert (z. B. durch `:hover`), beim Start und in Verbindung mit einer Transformation. Um einen weichen Übergang zu erzeugen, muss man den Namen (`transition-property`) der CSS Eigenschaft und die Dauer (`transition-duration`) angeben. Zusätzlich kann man noch eine Wartezeit (`transition-delay`) und die Geschwindigkeit (`transition-timing-function`) angeben.*

**CSS** `transition-property: [Wert];`



Als `[Wert]` gibt man entweder die CSS Eigenschaft an, oder `all` um allen Eigenschaften einen weichen Übergang zu geben.

`transition-property: background-color;`  
Der Übergang wird nur auf die Hintergrundfarbe des Selektors angewandt.

`transition-property: all;`  
Alle Eigenschaften des Selektors bekommen einen Übergang.

**CSS** `transition-duration: [Wert];`



Die Dauer des Überganges. Als `[Wert]` gibt man eine Zahl in Sekunden an.

`transition-duration: 2s;` 2 Sekunden

**CSS** `transition-timing-function: [Wert];`



Die Geschwindigkeit des Überganges.  
Als `[Wert]` sind unter anderen folgende möglich:

`ease` langsamer Start, dann schneller und gegen Ende wieder langsam  
`linear` Gleiche Geschwindigkeit  
`ease-in` langsamer Start  
`ease-out` langsames Ende

Darüber hinaus gibt es noch `steps(int,start|end)`, `step-start`, `step-end`, `ease-in-out` und `cubic-bezier(n,n,n,n)`.

**CSS** `transition-delay: [Wert];`



Verzögerung. Als `[Wert]` gibt man eine Zahl in Sekunden an.

`transition-delay: 3s;` 3 Sekunden Wartezeit bis zum Start.

```
.schalter {background:blue; color:white; padding:20px;
    transition-property:all; transition-duration:3s;
    transition-timing-function:ease;
    transition-delay:0.5s;}

.schalter:hover {background:yellow; color:black;}
```

**CSS** `transition: [property] [duration] [timing-function] [delay];`



Shorthand – man kann alle vier Eigenschaften mit einer zusammenfassen.

```
.schalter {transition: all 3s ease 0.5s;}
```



## Übungsblatt 13.6 Transitions

Referenzcode: HCU136

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
<b>Übung A</b>		
... erstellen Buttons mit weichem Übergang.	I	II
... erzeugen einen 3D Effekt mit einem Hover.	I	II
<b>Übung B</b>		
... erstellen ein CSS Accordion.	I	II
... beschäftigen sich mit dem Thema Drogen (Gesundheitsbildung).	SA	III
... öffnen und schließen anklickbare Elemente mit CSS.	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
<b>Übung C</b>		
... erstellen eine Webseite zum Thema "Mülltrennung".	I	II
... beschäftigen sich mit dem Thema "Mülltrennung" (Umwelterziehung).	SA	III
... gestalten die Webseite mit Tab-Reitern.	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

---



### Übung A: Button mit Übergang

- ☐ Erstelle ein neues HTML Dokument.
- ☐ Gestalte zwei Buttons mit weichem Übergang.
- ☐ Bei einem Hover soll ein 3D Effekt entstehen.



3D Effekte lassen einfach sich durch einen veränderten Box-Shadow realisieren.

**Zu unseren Angeboten**

**Zurück zur Website**



### Übung B: Accordion

- ☐ Erstelle ein neues HTML Dokument mit einem Accordion
- ☐ Die Kategorien sind:
  - Legale Drogen
  - Sanfte Drogen
  - Harte Drogen
- ☐ Suche im Internet nach Beispielen für die Kategorien (z. B: Harte Drogen, Heroin, Opium usw.) mit Wirkung und Gefahren.
- ☐ Durch einen Klick auf die Kategorie soll der Inhalt mit den Beispielen angezeigt werden – die anderen schließen sich.
- ☐ Das Öffnen und Schließen soll animiert sein.
- ☐ Gestalte die Webseite anspruchsvoll mit CSS.

**Legale Drogen**

**Weiche Drogen**

- Cannabis: wirkt psychoaktiv.
- LSD: kompletter Realitätsverlust.
- Ecstasy: kann Erektions- und Orgasmusstörungen auslösen.

**Harte Drogen**



### Übung C: Mülltrennung

- ☐ Erstelle eine Webseite zum Thema "Mülltrennung"
- ☐ Gestalte es mit einem Tab-Reiter.
- ☐ Freie kreative Übung.

## Lernhandout 14.1 CSS Variablen

Referenzcode: HCL141

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... erkennen und verstehen den Sinn von benutzerdefinierten Eigenschaften (custom properties) bzw. CSS Variablen.	I	I
... können Variablen festlegen.	I	II
... rufen Variablen innerhalb des CSS Bereichs ab.	I	II
... definieren ein Fallback für CSS Variablen, falls kein Browsersupport besteht.	I	II






**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

---

Mit benutzerdefinierten Eigenschaften (custom properties) oder CSS Variablen kann man zu Beginn Werte festlegen, die später bei einem Redesign nur einmal geändert werden müssen.

Browsersupport					
var()	49.0	15.0	31.0	9.1	36.0

## Variable festlegen

**CSS** `:root {--variablenname: WERT;}`



Die Variablen sollten zu Beginn der CSS Anweisungen definiert werden. Der Variablenname ist frei wählbar, man muss aber Groß- und Kleinschreibung beachten, weil diese Case-Sensitive sind.

```
:root {--akzent: #009;}
```

## Variable abrufen

**CSS** `css-selector {css-eigenschaft: var(--variablenname);}`



Die festgelegte Variable kann nun im gesamten CSS Bereich mit `var()` abgerufen werden.

```
h1 {color: var(--akzent);}
```



Um sicher zu gehen, kann man einen Fallback definieren der dann angewandt wird, wenn der Browser die Variablenfunktion nicht versteht. Dafür wird ein Beistrich gesetzt und der Fallback-Wert hinzugefügt.

```
h1 {color: var(--akzent, blue);}
```



## Beispiel zur Nutzung von unterschiedlichen Variablentypen

```
:root {--schriftfarbe: #333;
  --schriftart: Arial, Times;
  --schriftgr: 2em; }

h1 {color:var(--schriftfarbe, blue);
  font-family:var(--schriftart);
  font-size:var(--schriftgr, 3em);}

p {color:var(--schriftfarbe, black);
  font-family:var(--schriftart);}
```

---

```
<h1>Bahnfahren</h1>
<p>Gut und günstig - die Umwelt freuts!</p>
```

## Lernhandout 14.2 calc() Funktion

Referenzcode: HCL142

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... nutzen die CSS calc() Funktion um Berechnungen durchzuführen.	I	II
... kennen und nutzen die Operationsmöglichkeiten mit calc().	I	II
... achten dabei auf die korrekte Syntax.	IK	II
... kombinieren die calc() Funktion mit CSS Variablen var().	I	II
... definieren ein Fallback für Berechnungen.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene






**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

---

Mit der `calc()` Funktion können einfache Berechnungen in die CSS Anweisungen eingebunden werden. Diese sind jedoch nur bei Eigenschaften möglich, die ein Maß erwarten (z. B: `width`, `height`, `font-size`, `margin`, `padding` usw.). Dabei können Berechnungen zwischen unterschiedlichen Maßen angestellt werden

z. B. `width: calc(10em - 30px);`

Browsersupport					
<code>calc()</code>	26.0	9.0	16.0	7.0	15.0

### CSS Eigenschaft: `calc()`;



Mögliche Operationen sind

- Addition ( + )
- Subtraktion ( - )
- Multiplikation ( \* )
- Division ( / )

sowie Klammern. Vor und nach dem Rechenzeichen muss ein Leerzeichen gesetzt werden (zumindest bei Plus und Minus).

Unmögliche Rechenergebnisse (z. B: Division durch Null, oder eine negative Zahl für `width`) werden ignoriert.

```
font-size: calc(12pt + 1em);
width:      calc(80% - 2 * 3px);
min-width: calc((60% + 8px) / 2 );
```



Natürlich kann man die `calc()` Funktion mit CSS Variablen `var()` kombinieren.

```
:root {--breite:300px;
      --schrift:12pt;}

#eineBox {height: 200px; border: 3px solid red;
          width:   calc(var(--breite) + 100px);
          font-size: calc(var(--schrift) * 2);}

<div id="eineBox">Drinnen ist es dunkel</div>
```



Ein Fallback setzt man über die CSS Reihenfolge!

```
#eineBox {height: 200px; border: 3px solid red;
          width:   400px;
          width:   calc(var(--breite) + 100px);
          font-size: calc(var(--schrift) * 2);}
```



## Lernhandout 14.3 counter() Funktion

Referenzcode: HCL143

Feinziele Die Schüler innen ...	Zielart	Taxonomie
... nutzen die counter() Funktion um Elemente zu nummerieren.	I	II
... setzen den Zähler mit counter-reset auf Null.	I	II
... kombinieren die counter-increment Eigenschaft mit dem Pseudoelement ::before um eine Durchnummerierung vor einem Element zu erzeugen.	IK	II
... definieren das Aussehen und die Erscheinung einer Nummerierung im Pseudoelement ::before über die content-Eigenschaft und der counter() Funktion.	IK	II
... verschachteln Zähler über mehrere Ebenen.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

---

Neben dem `<ol>` Element, kann man auch mit CSS die Anzahl der Verwendungen durchnummerieren lassen. z. B. jede Überschrift `<h1>` durchnummerieren.

Dafür muss man zuerst den Zähler mit der CSS Eigenschaft `counter-reset` auf Null setzen und dem Zähler einen Namen geben. Vorzugsweise im Selektor für den `<body>`, damit der Zähler für das gesamte Dokument gilt.

**CSS** `counter-reset: NAME;`

```
body {counter-reset: abschnitt;}
```

Im Anschluss wird im Selektor für das gewünschte Element der Zähler aktiviert. Als Selektor verwenden wir ein `h2` Element mit einem `::before` Pseudoelement. Damit wird bei jeder Überschrift der zweiten Ebene vor dem Text der Zähler hinzugefügt.

**CSS** `counter-increment: NAME;`

```
h2::before {counter-increment: abschnitt;}
```

Mit der `content` Eigenschaft und der `counter()` Funktion bestimmen wir das Aussehen.

**CSS** `content: counter(NAME);`

```
h2::before {counter-increment: abschnitt;
             content: "Kapitel " counter(abschnitt) ": "};
```



### Zähler verschachteln (mehrere Ebenen durchnummerieren)

```
body {counter-reset: ersteEbene;}

h1 {counter-reset: zweiteEbene;}

h1::before {counter-increment: ersteEbene;
             content: "Kapitel " counter(ersteEbene) ": "};

h2::before {counter-increment: zweiteEbene;
             content: counter(ersteEbene) "." counter(zweiteEbene) " "};
```

```
<h1>Unternehmensführung</h1>
  <h2>Die Planung</h2>
  <h2>Die Kontrolle</h2>

<h1>Personalmanagement</h1>
  <h2>Arbeitnehmer haben Rechte</h2>
  <h2>Motivierte Mitarbeiter</h2>

<h1>Ökomanagement</h1>
  <h2>Grundlagen</h2>
  <h2>Instrumente</h2>
```

### Kapitel 1: Unternehmensführung

#### 1.1 Die Planung

#### 1.2 Die Kontrolle

### Kapitel 2: Personalmanagement

#### 2.1 Arbeitnehmer haben Rechte

#### 2.2 Motivierte Mitarbeiter

### Kapitel 3: Ökomanagement

#### 3.1 Grundlagen

#### 3.2 Instrumente


## Übungsblatt 14.3 CSS Funktionen

Referenzcode: HCU143

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... erstellen eine CSS Farbpalette mit Variablen.	I	II
... definieren die Farbvariablen in einem externen Stylesheet.	I	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
... binden das externe CSS Stylesheet in das Dokument ein und testen es mit beliebigen HTML Elementen.	I	IV
Übung B		
... definieren CSS Variablen für Schriftfamilien.	I	II
... definieren CSS Variablen für Schriftgrößen.	I	II
... erstellen Berechnungen mit calc() und den definierten Variablen nach schriftlicher Vorgabe.	I	III
... erstellen einen verschachtelten counter().	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

### Anmerkungen

-  Zu B: Es ist gesetzlich festgelegt, dass jede Webseite ein Impressum bzw. reale Kontaktdaten aufweisen soll. Das abrufbare Impressum (impressum.html) kann als Vorlage für andere online Webprojekte verwendet werden.



### Übung A: Farbpalette

- ☐ Erstelle ein neues CSS Stylesheet
- ☐ Definiere mit CSS Variablen eine eigene Farbpalette im externen Stylesheet. (`--meinBlau`, `--meinRot`, `--Akzent1` usw).
- ☐ Binde das CSS Stylesheet in ein beliebiges HTML Dokument ein. (TIPP: `<link rel="stylesheet" ...>`)
- ☐ Teste deine Farbpalette.
- ☐ Freie kreative Übung.



### Übung B: Impressum

- ☐ Öffne das HTML Dokument `impressum.html`
- ☐ Definiere die Schriftarten mit CSS Variablen.
  - Schriftfamilie für Überschriften
  - Schriftfamilie für Lauftext
- ☐ Definiere eine CSS Variable für die Schriftgröße mit 12pt.
- ☐ Die Schriftgrößen sollen mit `calc()` und der CSS Variable für die Schriftgröße berechnet werden.
  - `h1` = doppelt so groß wie die CSS Variable
  - `h2` = plus 0.5em zur CSS Variable
  - `h3` = plus 2pt zur CSS Variable
  - `p` = die CSS Variable
- ☐ Erstelle einen verschachtelten `counter()` für `<h2>` und `<h3>` (Nummerierung für die Überschriften)

## 1. Disclaimer – rechtliche Hinweise

### 1.1 Haftungsbeschränkung

Die Inhalte dieser Website werden mit größtmöglicher Sorgfalt erstellt. Der Anbieter übernimmt jedoch keine Gewähr für die Richtigkeit, Vollständigkeit und Aktualität der bereitgestellten Inhalte. Die Nutzung der Inhalte der Website erfolgt auf eigene Gefahr des Nutzers. Namentlich gekennzeichnete Beiträge geben die Meinung des jeweiligen Autors und nicht immer die Meinung des Anbieters wieder. Mit der reinen Nutzung der Website des Anbieters kommt keinerlei Vertragsverhältnis zwischen dem Nutzer und dem Anbieter zustande.

### 1.2 Externe Links

Diese Website enthält Verknüpfungen zu Websites Dritter ("externe Links"). Diese Websites unterliegen der Haftung der jeweiligen Betreiber. Der Anbieter hat bei der erstmaligen Verknüpfung der externen Links die fremden Inhalte daraufhin überprüft, ob etwaige Rechtsverstöße bestehen. Zu dem Zeitpunkt waren keine Rechtsverstöße ersichtlich. Der Anbieter hat keinerlei Einfluss auf die aktuelle und zukünftige Gestaltung und auf die Inhalte der verknüpften Seiten. Das Setzen von externen Links bedeutet nicht, dass sich der Anbieter die hinter dem Verweis oder Link liegenden Inhalte zu Eigen macht. Eine ständige Kontrolle der externen Links ist für den Anbieter ohne konkrete Hinweise auf Rechtsverstöße nicht zumutbar. Bei Kenntnis von Rechtsverstößen werden jedoch derartige externe Links unverzüglich gelöscht.

### 1.3 Urheber- und Leistungsschutzrechte

Die auf dieser Website veröffentlichten Inhalte unterliegen dem Urheber- und Leistungsschutzrecht. Jede vom Urheber- und Leistungsschutzrecht nicht zugelassene Verwertung bedarf der vorherigen schriftlichen Zustimmung des Anbieters oder jeweiligen Rechteinhabers. Dies gilt insbesondere für Vervielfältigung, Bearbeitung, Übersetzung, Einspeicherung, Verarbeitung bzw. Wiedergabe in Datenbanken oder anderen elektronischen Medien und Systemen. Inhalte und Rechte Dritter sind dabei als solche gekennzeichnet. Die unerlaubte Vervielfältigung oder Weitergabe einzelner Inhalte oder kompletter Seiten ist nicht gestattet und strafbar. Lediglich die Herstellung von Kopien und Downloads für den persönlichen, privaten und nicht kommerziellen Gebrauch ist erlaubt. Die Darstellung dieser Website in fremden Frames ist nur mit schriftlicher Erlaubnis zulässig.

## 2. Datenschutzerklärung

### 2.1 Datenschutz

Nachfolgend möchten wir Sie über unsere Datenschutzerklärung informieren. Sie finden hier Informationen über die Erhebung und Verwendung persönlicher Daten bei der Nutzung unserer Webseite. Ich weise ausdrücklich darauf hin, dass die Datenübertragung im Internet (z.B. bei der Kommunikation per E-Mail) Sicherheitslücken aufweisen und nicht lückenlos vor dem Zugriff durch Dritte geschützt werden kann. Die Verwendung der Kontaktdaten meines Impressums zur gewerblichen Werbung ist ausdrücklich nicht erwünscht, es sei denn ich hatte zuvor eine schriftliche Einwilligung erteilt oder es besteht bereits eine Geschäftsbeziehung.

### 2.2 Personenbezogene Daten

Sie können unsere Webseite ohne Angabe personenbezogener Daten besuchen. Soweit auf meinen Seiten personenbezogene Daten (z.B. Name, Adresse, E-Mail-Adresse) erhoben werden, erfolgt dies, soweit möglich, auf freiwilliger Basis. Diese Daten werden ohne Ihre ausdrückliche Zustimmung nicht an Dritte weitergegeben. Ich bin berechtigt, die personenbezogenen Daten zum Zweck der Vertragserfüllung zu verarbeiten. Die Verarbeitung ist rechtmäßig, wenn sie auf der Grundlage eines berechtigten Interesses begründet, inhaltlich ausgestaltet oder geändert werden sollen.

## Lernhandout 14.4 Attributselektoren

Referenzcode: HCL144

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... selektieren einzelne Elemente über ein bestimmtes Attribut.	I	II
... kennen die Syntax für Attributselektoren.	I	I
... erweitern einen Attributselektor mit einem zusätzlichen Wert.	I	II
... bestimmen Teilübereinstimmungen für Attributwerte (zu Beginn, am Ende und innerhalb).	IK	II
... kombinieren Attributselektoren mit anderen Selektoren.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene

**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

---

Mit dem einfachen Attributselektor können Elemente angesprochen, die ein bestimmtes Attribut besitzen. Zur Erinnerung: in `<p title="Absatz 1">` ist `title` das Attribut. Angesprochen wird das ganze Element indem der Attributselektor mit eckigen Klammern im CSS Bereich angegeben wird. z. B: `[title] {color:red;}`

#### CSS [Attribut]: {CSS Eigenschaft;}



Im Beispiel werden alle Pflichtfelder blau eingefärbt und alle Beschriftungen die ein Title Attribut haben eine rote Schriftfarbe zugewiesen.

```
[required] {background-color:blue;}
[title] {color:red;}
```

```
<label for="vorname" title="Vorname eingeben">Vorname: </label>
<input name="vorname" id="vorname" type="text" required ></p>

<label for="nachname" title="Nachname eingeben">Nachname: </label>
<input name="nachname" id="nachname" type="text" required ></p>
```

#### CSS [Attribut=Attributwert]: {CSS Eigenschaft;}



Gibt man dem Attribut zusätzlich (durch ein = Zeichen) einen Wert mit, so werden nur jene Elemente angesprochen, die sowohl Attribut als auch den passenden Wert haben. Im Beispiel wird nur der Reset-Button durchgestrichen.

```
[type=reset] {text-decoration:line-through;}
```

```
<button type="submit">Absenden</button>
<button type="reset">Zurücksetzen</button>
```



Es gibt drei Teilübereinstimmungen für die Attributwerte.

[Attribut^="abc"] ← Zeichenkette zu Beginn des Attributwertes  
 [Attribut\$="xyz"] ← Zeichenkette am Schluss des Attributwertes  
 [Attribut\*="klm"] ← Zeichenkette innerhalb des Attributwertes.

```
[href^="http://"] {font-weight: bold }
[href$=".com"] {color: green }
[title*="Graz"] {font-style:italic;}
```

```
<a href="http://www.css4.at">CSS4 Lernplattform</a>
<a href="http://www.wordpress.com">Blog Anbieter</a>
<h1 title="Für Graz und Graz-Umgebung">In ihrer Region</h1>
```



Attributselektoren können mit anderen Selektoren verbunden werden:

Universalselektor: `*[ Attribut]` → `*[title="Wichtig"] {...;}`  
 Type-selector: `element[Attribut]` → `a[href$=".net"] {...;}`  
 Klassenselektor: `.klasse[Attribut]` → `.bild[src*="png"] {...;}`



## Lernhandout 15.1 Image Map

Referenzcode: HCL151

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
... begreifen die Möglichkeiten von Image Maps.	K	III
... definieren eine Image Map mit dem Attribut usemap im <img> Element.	II	II
... verbinden Bild mit Image Map über den <map> Tag mit dem Attribut name.	I	II
... definieren den <area> Bereich und weisen die Attribute href (Hyperlink, Verweis) und target hinzu.	I	II
... bestimmen die Form des anklickbaren Bereichs mit dem shape Attribut.	I	II
... bestimmen die Koordinaten für den anklickbaren Bereich.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

Anmerkungen

---

Mit einer Image Map kann der Anwender per Maus auf ein Detail in einem Bild klicken um damit einen Verweis (Link) auszuführen. z. B. es wird eine Österreichkarte angezeigt und er User kann ein Bundesland auswählen. Zuerst wird ein beliebiges Bild mit dem Attribut `usemap` eingefügt (mit #). Dann ein `<map>` Element mit dem Attribut `name`, welches gleich sein muss (ohne #) wie das `usemap` Attribut im Bild. Innerhalb des `<map>` Tags wird über das `<area>` Element der anklickbare Bereich festgelegt.

**HTML** `<img src="" ... usemap="#mapname">`

```

```

**HTML** `<map name="#mapname"> ... </map>`

```
<map name="#AustriaKarte">
  <area shape="rect" alt="Wien" coords="694,132,738,168"
        href="http://www.wien.at">
</map>
```

**HTML** `<area shape="" coords="" href="" target="">`



Es können beliebig viele `<area>` Tags innerhalb des `<map>` Elements gesetzt werden. Folgende Attribute sind möglich:

`shape` bestimmt die Form

`shape="rect"` ← eine rechteckige Fläche

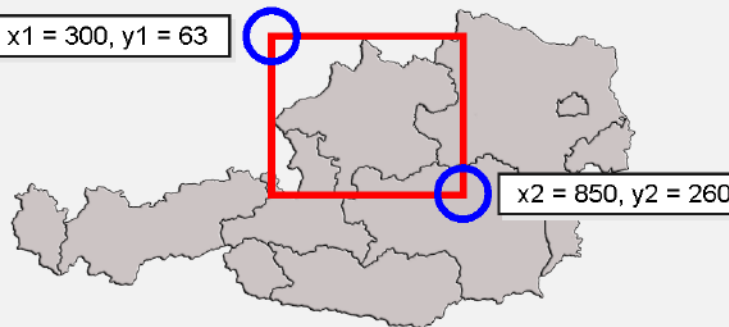
`shape="circle"` ← ein Kreis

`shape="poly"` ← ein Polygon

`coords` bestimmt die Koordinaten der Form (`<area>`).

Für `shape="rect"` gilt `coords="x1, y1, x2, y2"`

`x1 = 300, y1 = 63`



`x2 = 850, y2 = 260`

Für `shape="circle"` gilt `coords="x, y, r"`

`x` = Mittelpunkt, Anzahl der Pixel von links

`y` = Mittelpunkt, Anzahl der Pixel von oben

`r` = Radius in Pixel

Für `shape="poly"` gilt `coords="x1, y1, x2, y2 ... xn, yn"`

`href` und `target` benötigt man für den Verweis.

```
<area shape="rect" coords="340, 63, 580, 260"
      href="http://www.ooe.gv.at"
      alt="Oberoesterreich" target="blank" >
```

# Übungsblatt 15.1 Image Map

Referenzcode: HCU151

Feinziele	Zielart	Taxonomie
Die Schüler innen ...		
Übung A		
... nutzen ein Bildbearbeitungsprogramm (wie z. B. Photoshop oder Gimp) um in ein Bild fünf Fehler hinein zu retuschieren.	P	II
... binden beide Bilder (Original und Fehlerhaftes Bild) in ein HTML Dokument ein und erstellen eine Image Map über die eines der zwei Bilder.	I	II
... definieren den Area Bereich.	I	II
... kreieren ein Feedback (Rückmeldung) Modell.	IK	IV
... kombinieren die href Attribute mit einem Feedback für den User.	IK	II
... steigern ihr Selbstbewusstsein im Bereich Kreativität und Design durch freie Gestaltungsmöglichkeit.	A	II
... nutzen auch Image-Map Generatoren für die <map> und <area> Elemente.	I	II
... verändern mit CSS das Erscheinungsbild des Mauszeigers.	I	II

**Zielarten:** A ... Affektiv, K ... Kognitiv, P ... Psychomotorisch, S ... Selbstkompetenz, I ... Inhaltsebene  
**Taxonomien:** I = verstehen, II = anwenden, III = analysieren, IV = entwickeln

## Anmerkungen

- 📄 M. C. Eschers "Belvedere" aus dem Jahr 1958 ist wohl m. E. eines der schönsten Beispiele für optische Täuschung die im Surrealismus mündet.



### Übung A: Fehlersuchbild

- ☐ Öffne mit einem Bildbearbeitungsprogramm ein beliebiges Bild
- ☐ Retuschiere in der Bild fünf Fehler
- ☐ Erstelle ein neues HTML Dokument.
- ☐ Die fünf Fehler sollen über eine Image Map anklickbar sein.
- ☐ Baue ein Feedback ein  
(wenn man auf einen Fehler klickt, soll eine Rückmeldung kommen!)
- ☐ Gestalte die Seite mit CSS anspruchsvoll. Freie kreative Übung.



Im Internet findest du zahlreiche Image-Map Generatoren  
z. B. auf <https://www.image-map.net>



Mit der CSS Eigenschaft `cursor` kann man den Mauszeiger verändern.  
z. B. `#meinbild:hover {cursor:pointer;}`





# Pädagogischer Beipackzettel

## Warum überhaupt noch HTML?

Eine Frau Dr. Schubert und ihr Kollege Dr. Schwill haben eine äußerst erklärende Schrift mit dem Titel: "Didaktik der Informatik" herausgegeben. ISBN: 978-3-8274-2658-9

## Leistungsbewertung

Die Bewertung des|der Schüler|in ist keine interpersonelle sondern soll sich, getreu einem kognitivistischen Gesamtkonzept an der Prozessleistung des Lernerfolgs orientieren. Da die Leistungsbewertung im Vergleich zur Beurteilung immanent und ständig erfolgt, ist dem Lernprodukt nicht der gleiche Stellenwert beizumessen, wie es das punktuelle Lernprodukt einer Prüfungssituation nach sich zieht. So gesehen ist es klar, dass die Portfolio-Methode schlussendlich zur Beurteilung herangezogen werden kann – und m. E. auch in der Praxis passieren sollte. Die punktuelle Prüfung des Gelernten mag vielleicht einen Anreiz zur zweiten (bzw. mehrmaligen) Beschäftigung mit der Materie nach sich ziehen – ist aber m. E. nur sinnvoll in einem mündlichen Prüfungsgespräch, wo eigentlich nie die tatsächliche Festigung der Materie als eher die Fähigkeit Antworten auf Fragen zu geben geprüft wird. Ich empfehle, Kommentare im Quellcode des HTML und CSS Dokuments ebenfalls als prozessorientierte Leistung zu werten, wie das logisch-zielorientierte Denken – welches sich in der Klarheit eines EDV-Codes (in welcher Form auch immer) darstellt. Ein bedeutender prozessorientierter Akt im Webdesign liegt definitiv in der Formulierung von visuellen Vorgaben als auditiven Text (Code) und umgekehrt – die Antizipation des Codes zu seiner visuellen Darstellung. Man merkt, dass hier also linke und rechte Gehirnhälften gleichermaßen gefordert sind und sich wechselseitig bedingen.

## Blendet Learning

Im Informatikunterricht und seinen Verwandtschaftlich nahen Teilgebieten – ist der Begriff es blendet Learning wohl eher als ein leerer zu betrachten. Es führt zu einer paradoxen Unterrichtsgestaltung, sich selbst durch die Methode darzustellen – um so im Metabereich der Unterrichtspraxis auf das konzeptionell Höhere zurückzugreifen. Wie soll nun also das Begehren nach digitalen Kompetenzen und der Nutzung selbiger Bezugssysteme erfolgen? Die Loslösung erscheint für mich als das adäquate Mittel um das im System innenwohnende durch seine Vielseitigkeit als Prinzip aufzuführen. Die Substanz wird zum erkennenden Moment und entzieht sich einem geblendeten Lernen durch die Vielschichtigkeit ihrer eigenen Möglichkeiten. Weil nun das Rudimentäre zum Absoluten gesetzt wird, kann eine Begleiterecheinung eines Lernes nur durch die methodische Unterschiedlichkeit erklärt werden. Ich will wieder exemplarisch werden: Das Erlernen des Webdesigns kann sowohl über die vergleichbar simple Plattform [www.w3schools.com](http://www.w3schools.com) angeeignet werden – als auch über die als höher einzustufende Plattform [selhtml.org](http://selhtml.org). Die große (ultimative) Aufgabe dieser didaktischen Konzeption liegt im Mittelweg zwischen dem primitiv Einfachen und dem komplexen Vollständigen. Es liegt im Endeffekt an der Motivation des|der Schüler|in.

## Sozialformen

Aus pädagogischer Sicht bietet das weiterführende Turtoral keinerlei Anreize zu einem konsequenten Wechsel der Sozialformen. Da nun, wie es in den meisten IT dominierten Fächern der Fall ist, das Lernen von selbigen Inhalten die sich selbst bedingen, ein Lernen im isolierten Zustand – ein Lernen im reinen Ich nach-sich-zieht – kann kein großer Wechsel prognostiziert werden. In allen Fällen, selbst wenn von kolaborativen Lernen gesprochen wird, ist

reine Bildschirmarbeit weiterhin eine Isolation vom Wir-Gefühl aber: a) hochgradig potent um kognitive Ziele umzusetzen und b) eine Bereicherung des Ich-Gefühls solange sie als Selbstkompetenz verstanden wird und nicht einem kompetitiv-pädagogischen Szenario ausgesetzt wird. Man möchte meinen, es wird das Ich-Genie im Zusammenhang seines eigenen Denken konstruiert, dass sowohl dem Seienden seine Erwidierung Tribut leistet wie es im gemeinschaftlichen der kollektiven Erkenntnis jenes Moment liefert, dass noch fehlt. Im besten Fall könnte man ein Szenario der Bildschirmknappheit schaffen – man könnte den|die Schüler|in auffordern zu zweit an einer Konsole zu arbeiten – es ist aber nur eine Augen-auswischerei. Das Genie bleibt auch in diesem Bereich losgelöst von der Menge und erfährt damit kein besonders pädagogisches Moment – HTML und CSS ist nun mal nicht die Lösung aller Probleme des pädagogischen Alltags.

## Lehrplanbezug

Die genauere Darstellung eines Lehrplanbezuges ist im gegenwärtigen Zustand eher komplex. Natürlich würde ich eine Lehrstoffverteilung auf ein oder das nächste Semester hoch graduieren. Es entzieht sich jeglicher Basis – ein vorurteilendes Argument abzuliefern – weil nun der Lehrplanbezug in seinem Wesen mannigfaltig ist – und somit weitschweifig. Es gibt keine fundamentale Erkenntnis, die den Bezug in seiner endogenen Phase erkennt und so sein Wesen jeglicher esoterischen Basis entzieht. Ich habe schon einige gedankliche Versuche angestellt – um genau das nicht Obskure in den Denkprozess als ein Eliminat näher einfließen zu lassen. Die ganze Geschichte entbindet sich jeglicher Darstellungs-Objektivierung umso dem Gesamten seine Weißungskraft näher zu legen – wo wir uns fragen, warum auch nicht?

## Die kritische Sozialformierung

Der soziale Zusammenhalt im Schulischen ist definitiv nicht nur ein Fremdgesteuerter. Im Rahmen der Bildschirmarbeit, obgleich es in der allgemeinen Informatik oder deduktiv auf ein sich der informatisch gegenstandsbezogenen Unterrichtsdimension wie z. B. der Mathematik mit Hilfe von DO Kopfrechnen oder GeoGebra zur Darstellung erkenntnisrelevanter mathematischen und damit in Folge kognitiv/logischen Gedanken nach sich zieht. Tatsache ist, das jegliche Bildschirmarbeit immer eine Arbeit auf isolierter Basis ist. Auch die 'geblendete' Vorstellung einer IT-Tätigkeit über kollaborativen oder sozialen Netzwerken ist definitiv nicht (und ich wollte schon bei Gott) ein Ersatz zur Erlangung von kommunikativen oder sozialen Kompetenzen. Die Übung des logischen Denkens – der kommunikativen Formen und seiner Aktion-Reaktion, mag vielleicht simuliert sein und dem|der Lernenden ein Moment des Selbstbewusstseins verleihen, das sich nicht durch die konkrete als mehr durch die indirekte Konsequenz, abbildet und einstellt. Natürlich lassen sich so in der Sozialform der Einzelarbeit, kognitive Fehler in Form eines Syntax Errors oder eines Syntax Errors in Zeile xy dardarlegen – sie sind aber noch lange nicht die Fehlleistung des interaktiven in jedem menschlichen Prozess. Das isolierte Lernen zeigt paradoxerweise seine volle Wirksamkeit im perfekt-pädagogischen Wirken, dem geplanten und programmierten Unterricht, der endgültigen Absage von unlogischen Fragen und der klaren Antwort darauf: dem Syntax Error. Seine Antwort ist trivial aber ehrlich.



# Quellenverzeichnis

## Internetquellen und online Tools

Background Image Generator

<http://bg.siteorigin.com/>

css4.at Lernplattform

<https://www.css4.at>

dict.cc Wörterbücher

<https://www.dict.cc/>

Duden online

<https://www.duden.de/woerterbuch>

ExtractMetaData – Meta Daten auslesen

<https://www.extractmetadata.com/de.html>

flickr Fotocommunitiy

<https://www.flickr.com/>

Fundamentum – pädagogisches Forum

<http://fundamentum.xobor.de/>

Google (nahezu alle Dienste)

<https://www.google.at>

MDN web docs – Mozilla für Entwickler

<https://developer.mozilla.org/de/>

Mediaevent – CSS, HTML und JS mit {stil}

<https://www.mediaevent.de/>

openstreetmap

<https://www.openstreetmap.org>

Peter Kropff Tutorials

<https://www.peterkropff.de/>

pixabay Bilderplattform

<https://pixabay.com/de/>

selfhtml.org – Die Energie des Verstehens

<https://selfhtml.org/>

stackoverflow – developers empowering

<https://stackoverflow.com/>

W3C CSS Validation Service

<http://jigsaw.w3.org/css-validator>

W3C Markup Validation Service

<https://validator.w3.org>

w3schools.com – Web Developer Site

<https://www.w3schools.com/>

Wikimedia Commons

<https://commons.wikimedia.org>

Wikipedia – die freie Enzyklopädie

<https://de.wikipedia.org>

## Literaturverzeichnis

Adorno, Theodor W., Ästhetische Theorie, Suhrkamp Taschenbuch Verlag, Frankfurt am Main 1970

Bauer, Joachim, Lob der Schule, Heyne Verlag, Hamburg 2008

Doralt (Hrsg.), Werner, Kodex Schulgesetze, LexisNexis Verlag, Wien 2010

Esteras, Remacha, Infotech – English for computer users, Cambridge University Press 2008

Gaisbacher, Johann, Pongratz, Hanns Jörg (Hrsg.), Persönlichkeiten stärken, Leykam Graz, 2012

Glattauer, Niki, Der engagierte Lehrer und seine Feinde, Verlag Carl Ueberreuter, Wien 2010

Goleman, Daniel, Kaufman, Paul, Ray, Michael, Kreativität entdecken, Carl Hanser Verlag, München-Wien 1997

Goleman, Daniel, Soziale Intelligenz, Knaur, München 2008

Gruschka, Andreas, Verstehen lehren, Reclam Verlag, Stuttgart 2011

Gudjons, Herbert, Spielbuch Interaktions-Erziehung, Julius Klinkhardt, Bad Heilbrunn 2003

Hackenberg, Heide, Kommunikationsdesign Akquisition und Kalkulation, Verlag Hermann Schmidt, Mainz 2002

Heckhausen, Heinz, Hoffnung und Furcht in der Leistungsmotivation, Verlag Anton Hain, Meisenheim am Glan 1963

Hesse, Jürgen, Handbuch Vorstellungsgespräch, Eichborn Verlag, Frankfurt am Main 2001

Hüffel, Clemens, Reiter (Hrsg.), Anton, Handbuch Neue Medien, CDA Verlag, Perg 2008

Jacobsen, Jens, Website Konzeption, Addison-Wesley Verlag, München 2009

Jarosch-Frötscher, Carla, u. a., Businessstrainig, Übungsfirma, Case Studies V HAK, Trauner Verlag, Linz 2008

Jarz, Thorsten, Grundlagen der Netzwerktechnik, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2011

Jarz, Thorsten, VB.NET Eine Einführung ins Programmieren, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2010

Jarz, Thorsten, Visual Basic für Applications in Excel, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2009

Jarz, Thorsten, Windows 7, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2010

Jarz, Thorsten, Windows 8, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2013

Jarz, Thorsten, Windows Server 2008, Servicebetrieb ÖH-Uni Graz GmbH, Graz 2011

Jungmayer, J. R., Reproduktions und Druck Technik, Leykam-Verlag, Graz ISBN 3701112444

Klafki, Wolfgang, Neue Studien zur Bildungstheorie und Didaktik, Beltz Verlag, Basel 1985

Kölbl, Doris, Hutz, Gabriela, Lehrerinnenarbeit – heute und morgen, StudienVerlag GesmbH, Innsbruck 1997

Kopeinigg, Christine, Textdesign 2007 Informations- und Officemanagement 1/I, Wien 2007

Kopeinigg, Christine, Textdesign und Publishing, Informations- und Officemanagement 1/I, Wien 2011

Lauffer, Dora, Der Weg zum erfolgreichen Unterricht, Weishaupt Verlag, Graz 1996

Liessmann, Konrad Paul, Theorie der Unbildung, Piper Verlag, München-Wien 2011,

Mitschka, Ruth, Die Klasse als Team, 1997

Mitschka, Ruth, So lernt man lernen, 1983

Moriz, Werner, Unterrichtswissenschaften, Books on Demand GmbH, Norderstedt 2007

Pesendorfer Robert u. a. Informationsmanagement Office 2010 I/1, Trauner-Verlag Linz, 2013

Pesendorfer Robert u. a. Informationsmanagement Office 2010 II/2, Trauner-Verlag Linz, 2013

Pesendorfer Robert u. a. Informationsmanagement Office 2010 III/3, Trauner-Verlag Linz, 2013

Peterßen, Wilhelm H., Kleines Methoden-Lexikon, Oldenbourg 2009

Pflugfelder, Michael, Normativität und Professionalität in der Übungsfirma, Grin Verlag, Norderstedt 2007

Posch, Peter, 9 x Partizipation – Praxisbeispiele aus der Schule, Verlag Carl Ueberreuter, Wien 2006

Poser, Hans, Wissenschaftstheorie, Reclam Verlag, Stuttgart 2001

Punkenhofer, Yvonne, Das Funktionieren des Übungsfirmenmarktes, VDM Verlag Dr. Müller, Saarbrücken 2011

Schaub, Horst, Zenke, Karl, Wörterbuch Pädagogik, DTV München 2007

Schischkoff, Georgi, Philosophisches Wörterbuch, Kröner Verlag, Stuttgart 1991.

Schmidt, Siegfried J., Handbuch Werbung, LIT Verlag, Münster 2004

Schubert, Sigrid, Schwill, Andreas, Didaktik der Informatik, Spektrum Verlag Heidelberg 2011

Schulz-Reiss, Christine, Nachgefragt: Philosophie, Basiswissen zum Mitreden, Loewe Verlag GmbH, Bindlach 2005

Seeborn, Joachim, Kompakt-Lexikon Werbepaxis, Verlag Gabler GmbH, Wiesbaden 2001

Specht, Werner, Thonhauser Josef (Hrsg.), Schulqualität, Innsbruck 1996

Stephan, Ingrid, Büroprozesse in der Übungsfirma, Bildungsverlag EINS, Köln 2011

Stettner, Marko, Manipulation und Pädagogik, Leykam/Pädagogischer Verlag, Graz 1973

Watzlawick, Paul, Die erfundene Wirklichkeit, Piper Verlag, München 1981

Whitehead, Alfred North, Wissenschaft und moderne Welt, Suhrkamp Taschenbuch Verlag, Frankfurt am Main 1984

Winter, Felix, Grundlagen der Schulpädagogik Band 49: Leistungsbewertung, Schneider Verlag Hohengehren, 2012

Winterhoff, Michael, Thielen, Isabel, Persönlichkeiten statt Tyrannen, Wilhelm Goldmann Verlag, München 2011